논문 18-43-09-13 The Journal of Korean Institute of Communications and Information Sciences '18-09 Vol.43 No.09
https://doi.org/10.7840/kics.2018.43.9.1501

협력적 파워 프로파일 추적을 이용한 멈춤에 안전한 IoT 구동형 응용의 대규모 구성 방법

문 현 균◆, 조 정 훈*, 박 대 진°

Large-Scale Construction of Freeze-Safe IoT-Driven Applications Using Collaborative Power Profile Tracking

Hyeongyun Moon◆, Jeonghun Cho*, Daejin Park°

요 약

최근, IoT 시스템이 모든 일상생활에 적용되면서, 그에 따른 연결성과 수 많은 종류의 디바이스 때문에 그 시스템의 복잡성이 증가하고 있다. 또한 복잡하게 구성된 연결성 때문에 의도하지 않은 비정상적인 상태 전파에 매우 취약하다. 이 문제를 해결하기 위해 일반적으로 시스템 모니터링 회로를 추가하는데, 일반적인 모니터링 회로들은 모든 경우의 수에 대처하기 위해 과도한 스펙을 가지고 있다. 따라서 모니터링 회로에 의한 추가적인 전력 소모가 심하다. 본 논문에서는 이 문제를 해결하기 위해 통신 중심형 System-on-Chip (SoC)에서 임베디드 하드웨어와 소프트웨어의 저전력 동작을 위한 효과적인 전력 관련 파라미터 재구성 방법과 불규칙적인 IoT 디바이스들의 연결로 이루어진 대규모 구조 시뮬레이션 프레임워크 및 설계 공간 탐색 기법을 제안한다. 제안된 SoC 구조로 이루어진 대규모 구성 시뮬레이션 실험 결과, 매우 넓은 모니터링 주기로도 시스템을 안정성 있게 구동할 수 있었고, 이에 따른 전력 감소 효과도 얻을 수 있었다.

Key Words : Heterogeneous co-simulation, IoT system, Reconfigurable SoC, Low-power system, VPI, Freezing

ABSTRACT

The IoT systems, which are connected with irregular links between heterogeneous things, are vulnerable to fast error propagation caused by the unwanted abnormal status. To protect unresolved freezing due to this problem, we often add system monitoring circuits. However, practical approaches in field result in over-specification design to cover all unknown problems, so that a large amount of additional power is consumed. In this paper, we adopt two ways to efficiently solve this problem: the effective reconfiguration of power state parameters for the low power operations of embedded hardware and software in communication-centric system-on-chip (SoC), and design space exploration framework under the large-scale construction of irregular links between the processors. In case of large-scale connection of SoCs for the experiment, the system can be safely operated with a very wide monitoring active cycle, and the power reduction effect can also be obtained.

※ 본 논문은 교육부의 재원으로 한국연구재단의 기초과학연구 프로그램의 지원을 받아 수행된 연구결과임 (No. 2014R1A6A3A04059410 and 2016R1D1A1B03934343).

◆ First Author : (ORCID:0000-0002-0022-4739)Kyungpook National University, School of Electronics Engineering, moonhg1209@gmail.com, 학생회원

° Corresponding Author : (ORCID:0000-0002-5560-873X)Kyungpook National University, School of Electronics Engineering, boltanut@knu.ac.kr, 정회원

* Kyungpook National University, School of Electronics Engineering, jcho@knu.ac.kr, 정회원

논문번호 : 201806-0-077-SE, Received April 30, 2018; Revised August 26, 2018; Accepted August 27, 2018

Ⅰ. 서 론

1.1 서론

최근 IoT가 일상생활 대부분에 적용되면서, 하나의 시스템을 구성하고 있는 IoT 디바이스들의 종류도 다양해지고 개수도 많아지고 있다. 그리고 그림 1과 같이, 이러한 디바이스들이 상호 연결되어 동작을 수행하고 있기 때문에, 이에 따른 연결성도 점점 복잡해지고 있다. 그리고 이러한 시스템들이 모여서 또 하나의 큰 시스템을 구성하는 방식으로 점점 시스템의 복잡성도 증가되고 있다. 따라서 그림 2처럼, 하나의 디바이스에서 발생한 에러는 주변의 디바이스들에 영향을 주고, 이러한 에러들이 시스템 전체로 빠르게 전파되어 IoT 시스템 전체가 멈추거나 오동작 할 수 있다. 본 논문에서는 이러한 비정상적인 현상을 프리징(Freezing state)이라 정의하고 있다. 따라서 이러한 특성을 가지고 있는 IoT 시스템에서는 프리징을 막는 것이 매우 중요하다[1,2].

이러한 문제가 치명적으로 드러나는 두 가지 예를 보면, 첫 번째로 산업 전 분야에서 적용되고 있는 스마트 팩토리 (Smart factory)다[3]. 이러한 스마트 팩토리는 각각의 공정들이 자동화 되어있고, 이러한 자동화 된 공정들이 서로 유기적으로 동작하여, 전체 공정들이 인간의 개입을 최소화한 상태로 자동화되어 이루어진다. 이 때, 각 파트별로 나누어진 작은 공정 내부의 한 디바이스가 오동작 한다면, 이에 따른 영향은

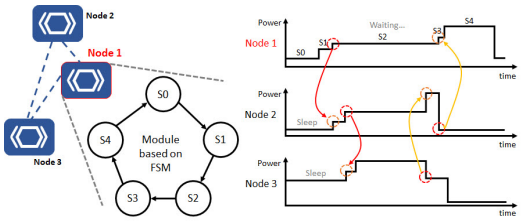


그림 1. 노드 간의 동작 원리 및 구조
Fig. 1. Principle and structure of operation between nodes

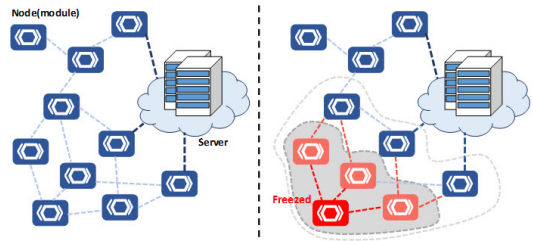


그림 2. 정상/비정상적 (Freezing)인 IoT 시스템
Fig. 2. Normal/Abnormal IoT system

해당 공정 파트에 영향을 주고, 시간이 지나면 공정 전체에 심각한 영향을 줄 것이다. 따라서 최대한 빠른 시간 내로 오동작을 검출하고 해결해야 전체 공정의 프리징을 막을 수 있다.

두 번째로, 분산 컴퓨팅 (Distributed computing) 분야에서 찾을 수 있다. 여러 독립된 디바이스 (노드)들이 네트워크를 통해 연결되어 막대한 양의 연산을 나누어 처리하는 방식에서의 분산 컴퓨팅은, 무엇보다 각 노드간의 유기적인 관계가 매우 중요하고, 각 노드의 연산결과들이 통합되어 큰 결과를 도출한다. 이 시스템에서도 역시 하나의 노드에서 오동작으로 인해 잘못된 연산을 하거나 결과 값을 송신하지 못할 경우에는 전체 결과에 문제를 주기도 한다. 따라서 최소 범위에서 오동작을 방지해야만 전체 결과의 무결성을 유지할 수 있다.

기존의 안정성 관점에서 IoT 시스템을 구성하는 방법이나 단일 디바이스에서의 저전력 솔루션들은 여러 제시되었었다[4,5]. 하지만 이미 제시된 방법들은 시스템에서 주고받는 데이터 관점에서 모니터링이 이루어지고 있고, 복잡성이 증가된 IoT 시스템에서의 모든 경우의 상황을 처리하기 위해 전력 소모 관점에서의 접근이 부족했다.

따라서 IoT 시스템을 안전하게 모니터링하기 위하여 과도한 스펙으로 모니터링 기능이 설계되었다. 하지만 이런 과도한 모니터링 회로는 그림 3(a)처럼 추가적인 전력 소모가 심해진다. 하지만 전력소모를 최소화하기 위해 설계한다면 그림 3(b)처럼 시스템 전체가 프리징 되는 현상을 막지 못 할 수도 있다. 따라서 그림 3(c)처럼 전력 소모를 최소화하고 시스템의 안정성을 보장할 수 있는 최적화된 모니터링 기능을 추가할 필요가 있다.

모니터링 회로의 추가적인 전력 소모는 전체 시스템의 전력 소모를 증가시킨다. 모니터링 회로의 전력 소모는 현재 MCU의 전력 소모를 측정하기 위한 고해상도의 ADC와, ADC를 통해 측정된 전력 소모와

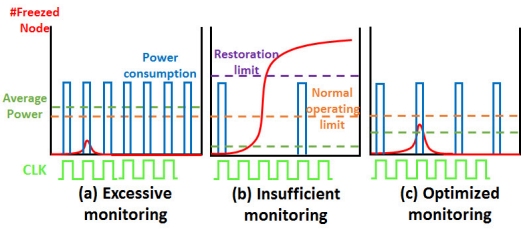


그림 3. 모니터링 주기에 따른 전력 소모 및 프리징 전파
Fig. 3. Power consumption and freezing propagation according to monitoring cycle

LUT에 저장된 정상상태의 예측된 전력소모를 비교하기 위한 비교기 등에서 발생한다. 이러한 전력 소모 특성 때문에 앞에서 언급했던 것처럼, 단일 디바이스 단위에서 최적화된 저전력 접근법들이 많이 소개되었다. 하지만 전체 시스템 단위에서 전력 관점에서 최적화하는 접근은 부족하였다.

전체 IoT 시스템의 전력소모는 식 (1)과 같이 표현될 수 있다. 그리고 모니터링 회로가 포함된 로직의 전력소모는 식 (2)로 표현될 수 있다. 식 (2)에서 해당 로직의 활성화 주기 A가 로직이 소모하는 전력의 큰 부분을 결정한다고 볼 수 있다. 따라서 모니터링 활성화 주기를 통해 감시 로직의 전력소모를 최소화 할 수 있고, 이를 통해 전체 시스템의 동작 전력도 절감될 수 있을 것이다.

$$P_{total} = \sum (P_{mcu} + P_{monitoring\ circuit} + P_{communication}) \quad (1)$$

$$P_{logic} = AfCV^2 + \tau AfVI_{short} + VI_{leak} \quad (2)$$

A: Activation period of a component
f: Clock frequency of the circuit
C: Capacitance of a circuit
V: Voltage in circuit
τ: Gate transition time
I_{short}: CMOS short-circuit current
I_{leak}: Leakage current

따라서 본 논문에서는 C/C++ 언어와 Verilog HDL을 Verilog Procedural Interface (VPI)를 통해 연동한 이기종 통합 시뮬레이션 구조 (Heterogeneous co-simulation platform)[6,7]와 멈춤에 안전하고 재구성 가능한 통신 중심의 IoT 구동형 SoC 구조를 제안한다. 제안된 시뮬레이션 구조를 이용하여 저전력과 시스템 안정성에 최적화된 모니터링 파라미터를 찾고, 이를 이용하여 적용될 시스템에 최적화 될 수 있게 제안된 SoC를 재구성 (Reconfiguration)한다. 제시된 구조에서 재구성된 모니터링 회로가 최적화된 파라미터를 이용하여 Micro-controller (MCU)의 전력소모를 감시하므로, 최소한의 전력 소모로 단일 디바이스의 안정성을 보장할 수 있고, 이를 바탕으로 전체 시스템의 안정성을 보장할 수 있다.

본 논문은 다음으로 구성된다. 1장 2절에서는 이기종 통합 시뮬레이션의 필요성을 소개하고, 2장에서는 본 논문에서 제안된 SoC 구조와 각 SoC를 재구성하기 위한 대규모 이기종 통합 시뮬레이션 구조 및 설계 공간 탐색 기법을 소개한다. 3장에서는 제시된 시뮬레이션 플랫폼과 SoC 구조를 이용한 실제 시뮬레이션 결과를 보여주고 이를 분석한 뒤 마지막 4장에서 결론을 맺도록 한다.

1.2 이기종 통합 시뮬레이션

이기종 통합 시뮬레이션 (Heterogeneous co-simulation)이란, C/C++와 같은 소프트웨어적인 언어와 Verilog HDL 같은 하드웨어적인 언어 같이 다른 프로그래밍 언어, 다른 시뮬레이션 엔진을 사용하여 진행하는 시뮬레이션을 말한다.

대부분의 시뮬레이션은 소프트웨어 또는 하드웨어 시뮬레이션 중 하나의 종류로만 이루어진다. 이러한 시뮬레이션을 동종 시뮬레이션 (Homogeneous simulation)이라 한다. 본 논문에서는 소프트웨어 또는 HDL 동종 시뮬레이션 방식을 채택하지 않고 소프트웨어/하드웨어 통합 시뮬레이션 기법을 사용하였다. 물론 소프트웨어 또는 하드웨어 단일 구조로 시스템을 묘사할 수 있고 시뮬레이션도 가능하다. 하지만 소프트웨어 (C, C++, Matlab, etc) 시뮬레이션은 순차적 실행이 된다. 실제 IoT 시스템은 수백 개, 수천 개의 디바이스들이 각각 동시에 자신의 동작을 수행하고, 연결된 인접 노드들과 통신하고 있다. 이러한 시스템을 소프트웨어적으로 시뮬레이션 한다면 각각의 디바이스들이 협력적으로 동시에 동작하는 것을 보지 못하는 단점이 있다. 이러한 단점을 보완하기 위해 하드웨어로 시뮬레이션 하면 동시에 각 디바이스들을 동작시킬 수 있다. 하지만 HDL로 수많은 디바이스들을 객체화 시켜 직접 입력과 출력을 연결해야 협력적인 동작이 가능하다. 개발자 입장에서 수천 개의 노드를 일일이 손으로 연결시켜 주는 일은 비효율 적이다.

따라서 수 많은 노드들을 자동으로 배치하고 연결시킬 수 있는 방법이 필요하기 때문에 이기종 통합 시뮬레이션 방법이 필요하다. 소프트웨어 레벨에서는 그래프 자료구조를 이용하여 효율적으로 각각의 노드들을 연결시킬 수 있고, 하드웨어 레벨에서는 각 디바이스들을 묘사하여 동시에 동작시킬 수 있다. 따라서 본 논문에서는 통합적인 시뮬레이션 방법을 제안하였고, 이를 이용해 확장성을 가지는 대규모 IoT 구조를 구성하여 시뮬레이션 할 수 있다.

Ⅱ. 본 론

2.1 제안하는 SoC 구조

본 논문에서 제안하는 SoC (Freeze-safe IoT-driven reconfigurable communication-centric SoC) 구조는 그림 4와 같다. SoC 내부는 크게 MCU와 모니터링 회로로 구성되어 있다.

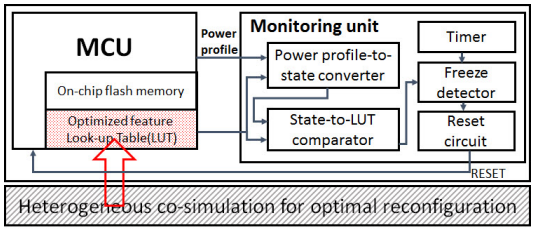


그림 4. 제안된 재구성 가능한 통신 중심형 SoC 구조
Fig. 4. Proposed reconfigurable communication -centric SoC architecture

MCU에서는 프로그래밍 된 코드에 따라 동작을 수행하고 온-칩 메모리에는 MCU의 특성이 저장되어 있는 룩-업 테이블 (Look-up table, LUT)이 저장되어 있다. 이 룩-업 테이블에 저장되는 정보는 그림 5와 같다.

MCU의 동작은 고정되어 있기 때문에, 인간의 지문같이 정상 동작일 때의 전력 소모 파형은 MCU별로 고유한 특성을 가진다. 따라서 정상 동작 상태의 전력 소모의 특징들을 저장하여 실시간 MCU 전력 소모와 비교를 통해 정상 동작 여부를 그림 6와 같이 감시한다. 기존에 예측된 정상 상태일 때 전력소모의 각 상태마다 시간과 실시간 전력 소모 파형의 각 상태별 시간이 허용 범위를 초과하여 일치하지 않는다면 프리즈 되었다고 판단한다. 이 때 LUT의 사이즈는 디바이스 종류와 저장된 정상동작 전력 소모 파형의 정밀도에 의해 결정된다. 디바이스 종류가 많아질수록, 전력 소모 파형을 더 정밀하게 관측할수록 LUT의 사이즈는 커진다. LUT의 데이터들은 제안된 시뮬레이션을 통하여 구해지고 저장된다. 이 저장된 데이터를 바탕으로 모니터링 회로가 재구성 된다. 모니터링 회로는 MCU의 실시간 전력소모를 측정하여 미리 정의된 상태로 바꾸는 전력 프로파일-상태 변환기, 상태로 매핑된 전력 소모를 LUT에 저장된 데이터와 비교하는 상태-LUT 비교기, 타이머를 기반으로 프리징 발생 여부를 감지하는 프리징 감지기, 프리징 발생 시 MCU를 리셋 시켜주는 리셋 회로로 구성된다. 이 때, 전력 프로파일-상태 변환기와 상태-LUT 변환기는 MCU에 저장된 LUT의 데이터에 따라 MCU에 맞게 재구성된다. 즉, 시뮬레이션을 통해 찾아진 모니터링 회로 활

Device	Active time	State 1	State 2	...	State M
Device 1	Optimized monitoring circuit active time	Timed features of the power profile in normal operations			
Device 2					
...					
Device N					

그림 5. 온-칩 플래쉬 메모리에 저장된 룩-업 테이블
Fig. 5. Look-up table stored in On-chip flash memory

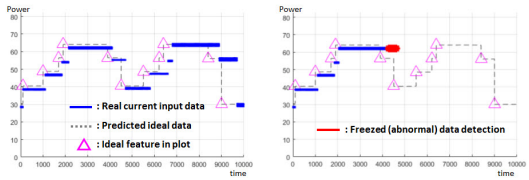


그림 6. 예측된 전력 소모를 기반으로한 프리징 감지
Fig. 6. Freezing detection based on predicted power consumption

성화 주기를 바탕으로 모니터링 회로가 재구성 되고, 결정된 주기마다 MCU의 전력 소모 변화를 관찰하게 된다.

2.2 제안하는 이기종 통합 시뮬레이션 플랫폼

최소한의 전력을 사용하여 MCU의 안전한 동작을 보장하기 위해서는 모니터링 회로의 감시 주기를 최적화해야 한다. 하지만 하나의 MCU는 여러 개의 노드들과 연동되어 동작하고, 이러한 수많은 노드들이 모여 IoT 시스템을 구성하고 있다. 본 논문에서는 이런 복잡한 시스템을 해석하고 최적의 활성화 주기를 찾기 위해 C 언어와 Verilog HDL 두 가지로 구성된 이기종 통합 시뮬레이션 플랫폼을 제안한다.

그림 7은 제안된 SoC 구조의 대규모 검증을 위한 이기종 통합 시뮬레이션 플랫폼의 전체적인 구조를 보여주고 있다. 시뮬레이션 플랫폼은 다른 프로그래밍 언어와 다른 시뮬레이션 엔진을 가지는 두 개의 계층으로 구성되어 있다. 첫 번째 계층은 C/C++ 언어로 구성되어 있고, 전체적인 IoT 시스템을 묘사하고 있다. 이 계층에서는 두 번째 계층과 통신하기 위한 소켓 서버와 IoT 시스템의 각 노드들의 연결성을 정의하기 위한 그래프 자료구조 모델로 구성되어 있다. 이 그래프 구조를 이용해 구성된 IoT 시스템은 각 노드 간의 실질적인 연결성만 가지고 있고, 해당 노드의 동작은 정의되어 있지 않다. 각 노드의 동작들은

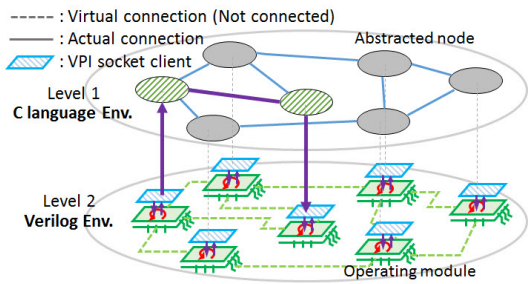


그림 7. 전체적인 통합 시뮬레이션 환경 구조
Fig. 7. Overall structure of co-simulation environment

Verilog HDL 기반으로 두 번째 계층에 모델링 되어 있다. 두 번째 계층은 IoT 시스템의 각 노드들의 동작이 정의되어 있는 수많은 모듈들과 각 모듈들이 자신의 상태를 소켓 통신을 이용하여 첫 번째 계층에 업데이트하기 위한 소켓 클라이언트로 구성되어 있다. 첫 번째 계층의 그래프 노드들과 두 번째 계층의 각각의 모듈들은 1:1로 매핑되어 있다. 두 번째 계층은 첫 번째 계층과 다르게 연결성은 전혀 정의되어있지 않고 각 디바이스의 동작만 정의되어 있다.

더 자세한 시뮬레이션 플랫폼 구조는 그림 8에서 볼 수 있다. C/C++ 환경에는 수많은 노드들이 연결되어 있는 IoT 시스템이 그래프 구조로 구성되어 있고, 각 노드들은 디바이스 종류를 나타내는 ID와 디바이스의 현재 상태와 이전 상태, 그리고 프리징 여부를 나타내는 Life 변수로 이루어진 구조체를 데이터로 가지고 있다. 소켓 서버를 이용하여 이러한 노드의 구조체 데이터들을 업데이트한다. 앞에서도 말했듯이 디바이스 동작 정의는 없고 디바이스 간의 연결성만 정의되어 있다. 각 노드의 데이터들은 두 번째 계층과의 통신으로 업데이트 되는데, 두 번째 계층은 Verilog HDL로 구성이 되어 있다. 각각의 노드들은 클락 단위로 동작하면서 자신의 상태를 첫 번째 계층의 자기 위치에 업데이트 하게 된다.

HDL 환경이 외부 환경과 통신하기 위해서는 Verilog Procedural Interface (VPI)라는 인터페이스를 이용해야 한다. VPI란, Verilog에서 C언어 같은 외부의 환경과 연동시키기 위해 사용하는 인터페이스인데, 그림 9처럼 Verilog 환경을 확장시켜주고 외부 환경으로의 통로를 열어주는 역할을 한다. 이 VPI를 사용해 소켓 클라이언트를 구성하여 C/C++ 환경과 통신한다. 각각의 모듈들은 자신의 상태를 소켓 클라이언트를 이용하여 첫 번째 계층으로 보내주고, 주변 노드들의 상태를 받아와 각각 동작한다. 두 번째 계층에서

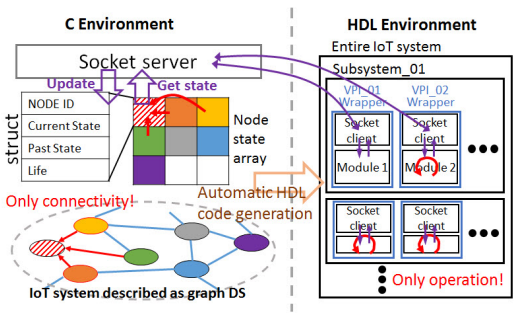


그림 8. 통합 시뮬레이션 플랫폼의 자세한 구성
Fig. 8. Detailed platform configuration of co- simulation environment

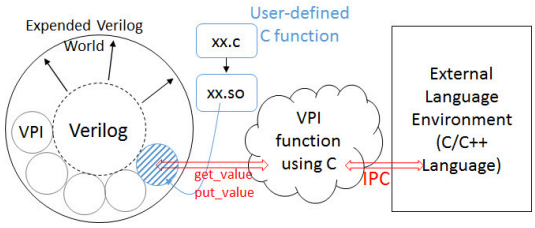


그림 9. Verilog procedural interface (VPI)
Fig. 9. Verilog procedural interface (VPI)

는 각 모듈의 동작만 정의되어 있을 뿐, 모듈 사이의 연결성은 전혀 정의되어 있지 않다.

본 논문에서 최적화된 활성화 주기를 찾기 위한 시뮬레이션 과정은 그림 10에서 볼 수 있다. 먼저 C/C++을 이용하여 그래프 구조를 기반으로 IoT 시스템을 구성하고, Verilog HDL을 기반으로 각 디바이스의 동작을 종류별로 정의한다. 앞서 구성된 그래프 구조의 IoT 시스템과 HDL로 모델링된 각 모듈들의 정보를 이용하여 두 번째 계층을 구성하는 수많은 모듈들을 HDL 코드로 자동 생성한다. 이러한 기법을 사용하여 HDL 환경을 자동 생성하지 않으면 수백, 수천 개의 노드를 HDL 레벨로 디자인 할 수 없다. 그리고 인위적인 에러 주입과 반복적인 설계 공간 탐색(Exhaustive design space exploration)을 이용한 시뮬레이션을 통해 최적의 모니터링 회로 활성화 주기를 찾는다. 이 때, 활성화 주기 조건으로 프리징 허용 범

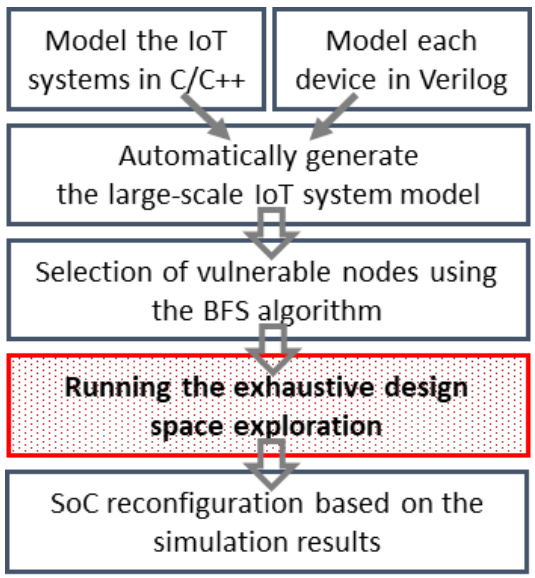


그림 10. 제안된 통합 시뮬레이션 과정 블록 다이어그램
Fig. 10. Proposed the co-simulation process block diagram

위를 지정한다. 전체 시스템의 10%까지는 프리징이 되어도 동작의 무결성을 유지하는 시스템이라면, 활성화 주기 조건으로 10%를 설정하면 된다. 그리고 찾은 주기를 MCU의 LUT에 저장하여 제안된 SoC를 재구성 한다.

IoT 시스템이 커질수록 시뮬레이션 시간도 비례하여 커지게 된다. 따라서 모든 노드에 대하여 시뮬레이션을 진행하기에는 무리가 있다. 여기서 그래프 자료구조의 장점이 드러난다. 모니터링 회로 활성화 주기는 시스템에서 최악의 상황에 맞춰 결정되기 때문에 가장 치명적인 노드에 대해서만 시뮬레이션을 진행하면 된다. 이런 취약한 노드를 찾기 위해 그림 11에서 설명되는 너비 우선 탐색 기법 (Breadth first search, BFS)이 적용된다. BFS를 해당 IoT 시스템에 적용하여 복잡한 연결성으로 인해 프리징 전파가 빠르게 일어날 노드를 찾은 뒤, 이 노드를 중심으로 시뮬레이션을 하면 시뮬레이션 시간을 효율적으로 단축시킬 수 있다.

따라서 제안된 시뮬레이션 플랫폼을 사용하면, 각 디바이스들의 동작만 HDL로 묘사할 수 있다면 노드 간의 연결성이 변경되어나, 노드 개수가 추가되더라도 첫 번째 계층에서 노드만 추가하여 효율적으로 시뮬레이션을 해볼 수 있다. 그리고 시뮬레이션을 통해 얻은 최적화된 모니터링 회로 활성화 주기를 이용하면, MCU도 프리징에 안전하게 동작할 수 있고, 모니터링 회로의 추가적인 전력소모도 최소화 될 것이다.

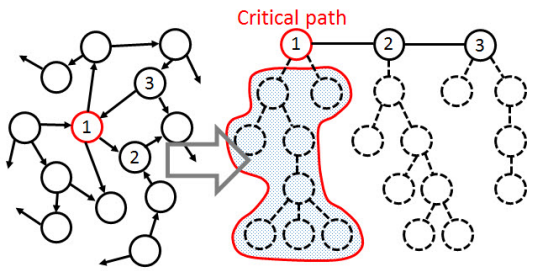


Fig. 11. Search breadth of each node in graph structure (Breadth First Search, BFS)
그림 11. 그래프 구조에서의 너비 우선 탐색

Ⅲ. 실 험

3.1 실험 환경

본 논문에서 제안되는 SoC 구조와 시뮬레이션 플랫폼을 실험 해보기 위해 IoT 시스템을 구현하여 시뮬레이션을 진행하였다. IoT 시스템은 450개의 노드를 가지고, 각 노드는 여러 개의 상태를 가지는 유한 상태 기계 (Finite state machine, FSM) 기반으로 동작하는 3 종류의 디바이스들로 구성되어 있다. 그리고 시스템에서 프리징 허용 범위를 10%로 지정하여 시뮬레이션을 진행한다. 임의의 시스템을 정의하기 위해, 450개의 노드 생성 후 랜덤 함수를 이용하여 노드들의 연결성을 정의하였고, BFS 알고리즘을 적용하여 각 디바이스 별, 가장 프리징에 취약한 노드를 선택하였다. 선택된 노드를 중심으로 에러를 주입하고, 모니터링 회로 활성화 주기를 변경해가며 시뮬레이션을 진행하였다. 이 모든 과정은 쉘 스크립터를 이용하여 자동화되어있다.

3.2 실험 결과

모니터링 회로의 활성화 주기에 따른 프리징된 노드의 최대 비율을 나타내는 그림 12에 나와 있다. 450개의 노드, 3 종류의 디바이스 중 복잡한 연결성으로 인해 프리징 전파가 가장 빠르게 발생할 노드 위주로 인위적인 에러를 주입하여 시뮬레이션을 진행하였다. 인위적인 에러란, 디바이스 동작이 중단되어 내부 클락 및 외부 인접 노드의 자극에도 상태 천이가 일어나지 않는 상태를 말한다. 시뮬레이션은 각 디바이스마다 활성화 주기를 변경하며 반복적으로 설계 공간 탐색을 실시하였다. 시뮬레이션 결과 활성화 주기가 넓어질수록 프리징 전파율도 커졌지만, 비례적으로 커지지는 않았다. 실험 환경 세팅에서 설정했던 프리징 전파 허용 한계점인 10% 지점에서 각 디바이스별로 활성화 주기를 선택하여 실제 시뮬레이션 상으로 구성된 IoT 시스템에 적용시켜 보았다. 이번 실험에 사용된 시스템에서의 디바이스별 최적화된 활성화 주기는 디바이스 1은 1415, 디바이스 2는 1230 그리고 디바이스 3은 1210이 된다. 단위 시간은 모니터링 회로 내

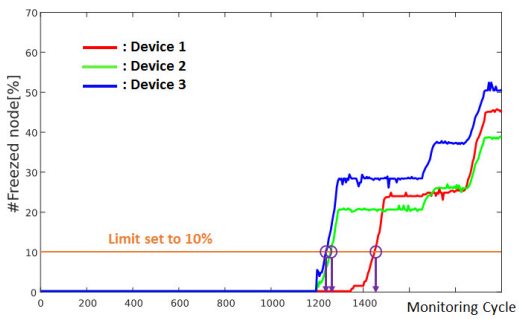


그림 12. 각 디바이스마다 모니터링 활성화 주기를 변경하며 진행한 시뮬레이션 결과
Fig. 12. Simulation result of each device according to parameter change

부의 타이머 기준으로 결정된다.

그림 13를 보면 제안된 시뮬레이션 플랫폼에서 얻어낸 최적화된 활성화 주기를 시뮬레이션 상에 구성된 IoT 시스템에 적용시킨 결과를 보여주고 있다. 450개의 노드 중 임의의 노드에 주기적으로 에러를 주입하여 프리징이 일어나는 것을 관찰하였다. 먼저 모니터링을 전혀 하지 않았을 때는, 전체 시스템의 약 60%까지 프리징이 일어나는 것을 볼 수 있었다. 하지만 최적화된 활성화 주기를 각 모듈에 적용하여 모니터링 회로가 MCU를 감시하였을 때는, 설정된 프리징 한계 범위 이내에서 프리징 전파를 차단하는 것을 볼 수 있었다. 또한 반복적 설계 공간 탐색을 이용하여 얻어낸 활성화 주기가 최적 값인지 증명하기 위해 찾아낸 주기보다 5% 큰 주기를 적용시켰을 때는, 프리징 한계를 초과하거나 시스템이 프리징으로부터 회복되는데 걸리는 시간이 늘어남을 볼 수 있다.

따라서 각 모듈들이 최적화된 모니터링 회로 활성화 주기를 바탕으로 동작하여, IoT 시스템의 전체적인 안정성을 보장할 수 있었고, 모니터링 회로의 활성화 빈도가 매우 낮아져 저전력 측면에서도 효과적인 방법이 될 것이다.

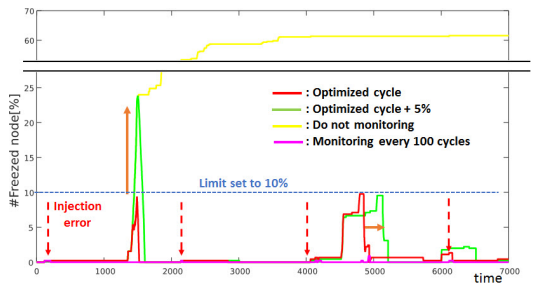


그림 13. 최적화된 모니터링 활성화 주기를 전체 시스템에 적용한 시뮬레이션 결과
Fig. 13. Simulation result of using optimized feature

Ⅳ. 결 론

본 논문에서는 제시된 이기종 통합 시뮬레이션 플랫폼 및 SoC 구조를 사용하여 특정 IoT 시스템의 저전력 동작 및 높은 안정성을 보장하는 방법을 제안했다. IoT 시스템의 상호 연결된 디바이스들의 전체적인 전력 소모 감소를 위하여, 모든 모듈들을 제안된 SoC 구조와 시뮬레이션을 통해 얻어진 특징점을 바탕으로 각각 재구성하였다. 통합 시뮬레이션 과정에서 수많은 각각의 모듈들을 클락 레벨로 정확히 구동하고, IoT 시스템의 특성인 복잡한 연결성 때문에 디바이스 레벨의 시뮬레이션에서는 무시되기 쉬운 디바이스 간의 상호 연결성까지 시뮬레이션 하였다. 따라서 시스템들의 시스템으로 구성되어 있는 IoT 시스템을 거시적인 관점에서 효과적으로 시뮬레이션 할 수 있었다. 시뮬레이션 결과를 보면 시스템의 특성에 따라 매우 넓은 간격으로 모니터링 회로를 작동시켜도 시스템의 안정성을 유지할 수 있었고, 감소된 모니터링 횟수만큼 전력 소모도 줄일 수 있다는 사실도 유추할 수 있다. 그리고 어떠한 시스템을 구성하더라도 각각의 디바이스들을 HDL로 묘사만 할 수 있다면, 디바이스 개수와 유형에 제한 없이 제안된 플랫폼에 적용하여 최적의 시스템을 구성할 수 있다. 따라서 여러 대규모 IoT 시스템에 효과적으로 적용할 수 있다.

References

[1] R. Roman, P. Najera, and J. Lopez, “Securing the internet of things,” *Computer*, vol. 44, no. 9, pp. 51-58, 2011.

[2] M. Diaz, C. Martin, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing,” *J. Netw. Comput. Appl.*, vol. 67, pp. 99-117, May 2016.

[3] P. Jadhav and I. G. Chun, “Fault detection and fault tolerant system for smart factories,” *2017 19th ICACT*, pp. 765-772, Bongpyeong, korea 2017.

[4] B. Song, “Low power solution for smart sensors in IoT,” *IEIE Mag. IEEE*, vol. 43, no. 1, pp. 24-31, 2016.

[5] W. Lyons, “Enabling increased safety with fault robustness in microcontroller applications,” *ARM Corporation*, 2010.

[6] J. Rowson, “Hardware/Software co-simulation,” in *DAC*, pp. 439-440, San Diego, California, 1994.

[7] F. Fummi, M. Poncino, S. Martini, F. Ricciato, G. Perbellini, and M. Turolla, “Heterogeneous co-simulation of networked embedded systems” in *Proc. DATE’04*, vol. 3, pp. 168-173, Paris, France, 2004.

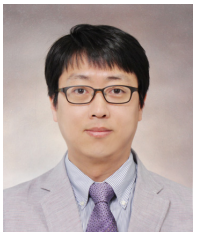
문 현 균 (Hyeongyun Moon)

2018년 2월 : 경북대학교 전자공학부 졸업
2018년 3월~현재 : 경북대학교 전자공학부 석사과정
<관심분야> 저전력 SoC 설계, 하드웨어-소프트웨어 Co-design

조 정 훈 (Jeonghun Cho)

1996년 : KAIST B.S
1998년 : KAIST M.S
2003년 : KAIST Ph.D
2003년~2005년 : 하이닉스 반도체 선임연구원
2005년~현재: 경북대학교 전자공학부 교수
<관심분야> 임베디드시스템, 바이너리 변환, 차량용 안전 및 보안시스템, AUTOSAR, 런타임 감시

박 대 진 (Daejin Park)



2001년 : 경북대학교 전자전기공학부 학사
2003년 : KAIST 전기 및 전자공학과 석사
2014년 : KAIST 전기 및 전자공학과 박사
2003년~2014년 : SK Hynix/Samsung (차세대 LSI 설계) 수석연구원
2014년~2016년 : 경북대학교 전자공학부 초빙교수 (2014년 대통령 Postdoctoral Fellow 선정)
2016년~현재 : 경북대학교 전자공학부 조교수
<관심분야> 전력 SoC 설계, 하드웨어-소프트웨어 Co-design, Dependable 스마트 IoT 시스템, Robust 임베디드 시스템