

# 멀티 코어와 GPU 결합 구조를 이용한 DEVS 기반 대규모 하이브리드 시스템 모델링 시물레이션의 가속화

김성섭 · 조정훈 · 박대진<sup>†</sup>

## Accelerated Large-Scale Simulation on DEVS based Hybrid System using Collaborative Computation on Multi-Cores and GPUs

Seongseop Kim · Jeonghun Cho · Daejin Park<sup>†</sup>

### ABSTRACT

Discrete event system specification (DEVS) has been used in many simulations including hybrid systems featuring both discrete and continuous behavior that require a lot of time to get results. Therefore, in this study, we proposed the acceleration of a DEVS-based hybrid system simulation using multi-cores and GPUs tightly coupled computing. We analyzed the proposed heterogeneous computing of the simulation in terms of the configuration of the target device, changing simulation parameters, and power consumption for efficient simulation. The result revealed that the proposed architecture offers an advantage for high-performance simulation in terms of execution time, although more power consumption is required. With these results, we discovered that our approach is applicable in hybrid system simulation, and we demonstrated the possibility of optimized hardware distribution in terms of power consumption versus execution time via experiments in the proposed architecture.

**Key words** : Discrete event simulation, hybrid simulation, GPU based accelerated simulation, simulation kernel distribution on multi-core architecture, M&S

### 요약

이산 사건 시스템 명세 (DEVS)를 이용한 하이브리드 시스템 시물레이션은 IoT 기반 Smart factory의 최적 동작을 위한 파라미터 추출 등 멀티 레벨 모델 계층을 포함한 복잡계 시스템의 해석에서 중요한 도구로 사용되고 있다. 하이브리드 시스템은 연속 시간 시스템과 이산 사건 시스템의 특성을 모두 포함하고 있어 그 복잡성으로 인해 결과를 얻기 위해 많은 시간을 필요로 한다. 본 연구에서는 멀티 코어와 GPU가 결합된 이기종 컴퓨터 구조를 이용한 DEVS 기반 대규모 하이브리드 시스템 시물레이션의 가속화를 제안한다. 제안하는 멀티 코어-GPU 상호 결합 시물레이션 실행 플랫폼을 사용하여 상대적으로 많은 순간 전력을 소모하지만 실행 시간 측면에서 빠른 시물레이션이 오히려 전체 에너지 소모 측면에서 장점을 가지는 것을 보여주고자 한다. 이를 위해 대규모 모델의 수평적/수직적 상호 결합된 DEVS 기반 하이브리드 시스템을 시물레이션 하였고 효과적인 시물레이션을 위한 하드웨어의 조합, 동작 파라미터 변경에 따른 성능 향상을 전력 소모 관점에서 분석하였다.

**주요어** : 이산 사건 시물레이션, 하이브리드 시물레이션, GPU 기반 가속 시물레이션, 멀티 코어 기반 시물레이션 커널 분배, M&S

\* 이 연구는 교육부의 재원으로 한국연구재단의 기초과학 연구 프로그램(No. 2014R1A6A3A04059410)의 지원을 받아 수행되었습니다.

**Received:** 6 June 2018, **Revised:** 8 August 2018,  
**Accepted:** 29 August 2018

**† Corresponding Author:** Daejin Park  
E-mail: boltanut@knu.ac.kr  
School of Electronics Engineering, Kyungpook National University

## 1. 서론

이벤트 기반 모델링 및 시물레이션은 사물인터넷과 같이 대규모 연결로 구성된 각각의 모델 구조를 다양한 계층으로 표현할 수 있고 시스템 간에 복잡한 상호관계와 그로 인한 상태변화의 동특성을 이벤트 기반으로 해석할 수 있어 대상 시스템의 복잡한 동특성을 쉽고 강력하게

분석할 수 있다. DEVS는 계층적이고 객체 지향적인 특성에 기반 하여 수평적으로 상호 결합된 시스템 구조와 수직적으로 멀티 레벨 모델링을 용이하게 기술하고 통합할 수 있는 형식론이며 이산 사건 시스템뿐만 아니라 연속 시간 시스템과의 이벤트 기반 연결성 등을 모델링하고 통합 시뮬레이션 하는 데 적용될 수 있는 방법이다.

특히 이벤트를 이용한 시뮬레이션의 시간 천이 동작으로 인해 고정시간 간격을 기반으로 하는 시뮬레이션에 비해 시스템 모델을 평가할 때 고속으로 시뮬레이션 할 수 있는 장점이 있다. 일반적으로 하이브리드 시스템은 두 가지 이상의 시스템의 특성들이 결합되어 있는 시스템을 의미하며, 본 연구에서 다루는 하이브리드 시스템은 연속 상태 특성과 이산 상태 특성 모두가 하나의 시스템 모델에 내재되어 있고 이를 단일 시뮬레이터로 해석하는 것을 가정한다. 연속 시간 시스템과 이산 사건 시스템의 결합성 때문에 여러 관점의 동특성을 가지는 복잡한 시스템의 모델링과 시뮬레이션에 효과적으로 이용될 수 있다.

본 연구에서는 DEVS 형식론 기반의 이산 사건 시스템과 연속 시간 시스템의 결합으로 구성된 하이브리드 시스템의 통합 시뮬레이션을 위한 이전 연구에 기반 한다. DEVS 기반의 하이브리드 시스템 모델링 관련 연구가 진행되었으며(Zeigler, 1995), 많은 분야에서 복잡한 시스템을 시뮬레이션 하는 데 이용되어 왔다(Bergero, 2011; Hong, 2013; Saadawi, 2012). 대부분의 경우에서 하이브리드 시스템은 수학적 기초를 기반으로 모델링된 연속 시간 시스템을 포함하므로, 이벤트 기반의 이산 사건 시스템 시뮬레이션에 비해서 해석이 복잡하고 통합 시뮬레이션에 많은 시간이 소요된다.

본 연구에서 모델링하고자 하는 시스템의 대상의 구조는 DEVS 형식론 기반으로 표현될 수 있는 셀룰러 모델(Wainer, 2004)을 이용하여 이산 시간, 공간, 그리고 시스템의 상태를 이용한 셀들의 대규모 상호 연결 형태의 집합으로 정의되어 많은 수의 모델의 셀들을 필요로 한다. 이러한 이유로 인해 속도 측면에서 보다 높은 성능의 시뮬레이션을 가능하게 하기 위해 시뮬레이션의 병렬화 방법을 고안하고 이를 하드웨어 플랫폼에 최적화 하는 것이 필수적이다. 이와 관련하여 빠른 시뮬레이션 처리 시간을 위해 시뮬레이션의 병렬 처리와 관련된 많은 연구들이 진행되었다(Liu, 2007). 특히 멀티 코어에서의 DEVS 형식론 기반의 시뮬레이션 연구와 GPU를 이용한 병렬 시뮬레이션 등의 연구들이 진행되었다(Liu, 2012). GPU는 많은 양의 연산에서 강점을 가지는 장치로서, CPU와 같이 사용되는 이기종 아키텍처로 많은 연구에서

사용되었다(Seok, 2012).

본 연구에서는 연속 시간 시스템과 이산 사건 시스템의 결합으로 이루어진 DEVS 형식론 기반의 하이브리드 시스템의 시뮬레이션을 가속화 관점에서 접근하였다. 특히, 대규모 셀룰러 기반의 하이브리드 시스템 시뮬레이션의 동작 과정을 CPU 멀티 코어와 GPU에 적절하게 분배하고, GPU를 이용한 대규모 셀룰러 모델의 시뮬레이션에서 고려해야 할 부분들을 제안하고자 한다. 제안하는 멀티 코어와 GPU가 결합된 이기종 아키텍처에서의 시뮬레이션을 시스템의 매개 변수들의 값을 바꾸어가며 결과를 분석하였다. 또한, 시뮬레이션에 소요되는 시간을 코어 개수에 따른 다양한 하드웨어의 조합에서 분석하였다. 더 나아가, 각각의 하드웨어 조합에서 시뮬레이션 시 소모되는 최대 전력 값을 측정하고, 제안하는 이기종 아키텍처에서의 시뮬레이션을 효율성 관점에서 분석하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 소개하며, 3장에서는 DEVS 형식론의 배경 지식을 설명한다. 4장에서는 제안하는 멀티 코어-GPU 기반의 이기종 아키텍처에서의 시뮬레이션 구조를 설명하며, 5장에서는 제안하는 시뮬레이션의 동작 과정과 타겟 시스템인 화재 모델을 설명한다. 실험 결과를 나타내는 6장에서는 제안하는 구조를 여러 하드웨어 조합에서 동작시키고, 시뮬레이션 시간, 전력 소모 관점에서 분석한다. 그리고 마지막 7장에서 결론을 맺는다.

## 2. 관련 연구

DEVS 형식론 기반의 하이브리드 시스템을 이용한 복잡한 시스템의 시뮬레이션에서 대상 모델을 부분 모델로 나누는 연구를 포함한 병렬 처리 연구들이 진행되었다(Chow, 1994; Jaer, 2009; Himmelspach, 2007). 고속 시뮬레이션을 위하여 DEVS 기반의 시뮬레이션을 병렬 처리하는 연구들이 꾸준히 진행 되었고(Troccoli, 2003; Chow, 1994), 가속기로 GPU를 이용하여 시뮬레이션 시간을 가속화하기도 하였다(Seok, 2012; Park, 2010). DEVS 기반의 시뮬레이션에서 타겟 시스템으로 셀룰러 모델을 이용하여 많은 연구들이 진행 되었으며(Wainer, 2001; 2016), 특히 화재와 같은 전파 현상을 모델링하고 시뮬레이션 하는 연구들이 진행 되었다(Ntaimo, 2004; Muzy, 2002). 일반적으로 셀룰러 모델로 이루어진 시스템의 시뮬레이션에서, 전체 시스템을 이루는 셀의 개수가 많을수록 더 상세한 시뮬레이션 결과가 얻어지지만, 반면 셀의 개수가 늘어날수록 더 많은 연산 양이 요구된다. 이러한

문제점을 해결하기 위해, 대규모 셀룰러 모델의 시뮬레이션 가속화 관점에서의 연구가 진행되었다(Hu, 2004). 본 연구는 기존의 GPU 기반의 DEVS 병렬 처리 연구와 달리, DEVS 기반의 하이브리드 시스템의 가속화된 시뮬레이션을 제안하고, 제안하는 이기종 아키텍처에서의 시뮬레이션의 성능을 상세한 시뮬레이션 결과를 얻을 수 있는 대규모 셀룰러 모델을 이용하여 분석하고 검증하였다.

본 논문에서는 DEVS 기반의 가속화 연구를 이기종 아키텍처가 결합된 구조의 구현 관점에서 진행한다. 특히, CPU와 GPU가 결합된 구조에서 연속 시간 시스템과 이산 사건 시스템이 결합된 하이브리드 시스템의 시뮬레이션을 가속화하고자 한다. 이를 위하여 시뮬레이션의 처리 과정을 각각의 하드웨어에 적절히 분배하고, CPU 코어와 GPU 코어 개수에 따른 여러 하드웨어 조합에서 시뮬레이션을 진행하고, 전력 소모를 분석하여 제안하는 구조의 효율성을 분석하고자 한다.

### 3. 배경 지식

#### 3.1 이산 사건 시스템

이산 사건 시스템은 시간 사건 시스템으로, 각각의 이산 상태가 시간에 따라 비동기적으로 발생하는 이벤트에 의해 갱신된다. 여러 모델링 방법이 시뮬레이션이 진행되는 과정 중에 동적으로 바뀌는 상태변화를 이산 사건 시스템의 형태로 모델링하기 위하여 사용되고 있다. 그 중에서, DEVS 형식론은 모듈화와 계층화된 구조로, 이산 사건에 따른 상태천이를 따르는 시스템과 미분 방정식으로 표현되는 연속 시간 시스템을 한 번에 표현할 수 있도록 하이브리드 시스템의 모델링으로 확장될 수 있다. DEVS는 또한 객체 지향적인 모델링 및 그에 따른 분석이 용이하기 때문에, DEVS를 이용한 실제 시스템의 모델링은 그 계층적인 구조로부터 원자 모델(atomic model)과 결합 모델(coupled model)로 표현될 수 있다. 이 중, 계층적인 구조에서 가장 하위의 기본적인 모듈인 원자 모델은 Figure 1과 같이 나타난다.

Figure 1은 이산 사건 시스템에 대한 DEVS 형식론의 원자 모델을 나타내며 다음은 해당 모델의 수학적 표현을 나타낸다. DEVS 형식론 기반의 원자 모델은 기본적으로 세 개의 변수와 네 개의 함수로 구성된다. 세 개의 변수는  $X, Y, S$ 로 나타나며 네 개의 함수는 외부 상태 천이 함수, 내부 상태 천이 함수, 출력 함수, 그리고 시간 진행 함수로 구성된다.  $X$ 는 이산 사건 시스템에서의 입력들의 집합을 나타내며,  $Y$ 는 이산 사건 시스템에서의

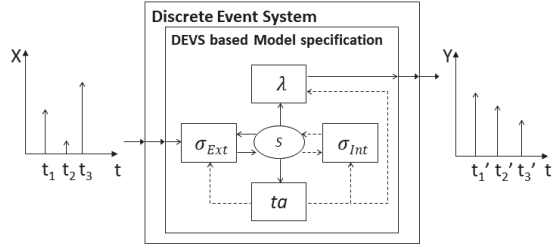


Fig. 1. Specification of discrete event system

출력의 집합을 나타낸다.  $S$ 는 이산 사건 시스템에서의 상태들의 집합을 나타내며, 이 상태 값들은 시스템의 모든 상태들을 나타낸다.  $\delta_{ext}$ 는 외부 상태 천이 함수를 나타내며 상태  $S$ 와 입력  $X$ , 그리고  $e$ 의 값에 영향을 받는다.  $e$ 는 마지막 상태 천이로부터 흐른 시간 값을 가진다.  $\delta_{int}$ 는 내부 상태천이 함수이며,  $\lambda$ 는 시스템의 출력 함수이다.  $ta$ 는 시간 진행 함수이며, 내부 상태 천이 함수의 트리거 동작 역할을 한다. 이산 값을 가지는 상태 변수들은 시스템이 외부에서 다른 시스템으로부터 입력을 받았을 때 변하게 된다.

만약 입력이 특정 시간 동안 들어오지 않는다면, 내부 상태 천이 함수가 미리 정해진 일정 시간이 흐른 후에 동작하게 되고, 그로 인해 상태가 업데이트 된다. 외부로부터 입력이 오기 전까지 기다리는 시간은 시간 진행 함수에 의해서 결정된다. 다른 시스템 모델링 방법과 달리 DEVS 모델링 시스템은 외부 입력에 의해서만 시스템의 상태가 변하는 것이 아니라 내부 상태 천이에 의해서도 시스템의 상태가 변하는 구조를 가지고 있다.

$$M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

$X$  : 이산 사건 입력 집합

$Y$  : 이산 사건 출력 집합

$S$  : 이산 사건 상태의 집합

$\delta_{ext} : Q \times X \rightarrow S$  : 외부 상태 천이 함수

$Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$

$\delta_{int} : S \rightarrow S$  : 내부 상태 천이 함수

$\lambda : S \rightarrow Y$  : 출력 함수

$ta : S \rightarrow \mathbf{R}_{0, \infty}^+$  : 시간 진행 함수

#### 3.2 연속 시간 시스템

연속 시간 시스템의 정의는 다음과 같다. 입력은 시스템의 상태에 영향을 주며, 그 상태는 현재 시점의 입력에 의해서만 영향 받는 것이 아니라, 과거의 상태와 입력에 의해서도 영향을 받는 미분 방정식 형태를 가진다. 미분

방정식의 기본 구조는 Figure 2의 연속 시스템 모델과 같이 나타난다. 미분 방정식을 풀기 위한 여러 방법 중, 본 연구에서는 Runge-Kutta 방법을 이용한다. Runge-Kutta 방법은 미분 방정식 중 초기값 문제를 해결하는 방법이며, 연산을 위한 시간 간격이 짧을수록 그 결과가 정확하다. 즉, 연속 시간 시스템 시뮬레이션 과정의 시뮬레이션 시간은 조밀한 많은 구간으로 나누어지며, 그로 인해 이산 시간 시스템의 시간 천이 및 상태 변환 특성을 따르게 된다. 시스템 모델은 시간 흐름에 의한 미분 방정식에 의해 영향 받고 업데이트 된다. 이러한 이산 시간 시뮬레이션의 특성 때문에, 이산 시간 시스템 모델은 연속 시간 시스템에 적용할 수 있는 시스템이다.

### 3.3 하이브리드 시스템

하이브리드 시스템은 이산 사건 시스템과 연속 시간 시스템의 결합된 시스템으로 나타난다. 이산 시간 시스템으로 모델링된 연속 시간 시스템은 미분 방정식  $\phi$ 의 집합으로 정의된다. 하이브리드 시스템은 미분 방정식에 의해 영향을 받는 연속 시간 시스템의 상태  $Q_{CS}$ 를 가진다. 하지만, 시뮬레이션 과정 중 시스템이 외부로부터 받는 입력은 시간 함수에 의한 이산 값이다. 즉, 이산 값을 변환하는 과정이 필요하며, 그 과정은 이벤트를 연속 상태로 변환하는 과정을 포함한다. 이러한 동작을 만족시키기 위하여, 시스템은 이벤트에서 연속 상태로 변환하기 위한 임계값을 동시에 고려하여 모델링되어야 한다. 임계값은 특정 시간 시점에서 특정 상태  $f_i$ 에 도달했을 때의 상태를 말하며, 그 임계값은 모델링 과정 중에서 정의된다. 연속 시간 시스템과 이산 사건 시스템을 결합한 하이브리드 시스템은 Figure 2와 같다.

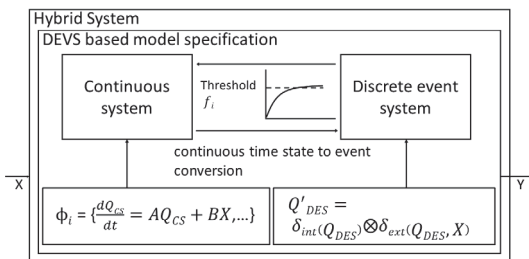


Fig. 2. Specification of hybrid system

## 4. 제안하는 구조

### 4.1 멀티 코어-GPU 결합 시뮬레이션

멀티 코어와 GPU가 결합된 구조의 대규모 하이브리드

드 시스템의 시뮬레이션 가속화를 수행하기 위해 각각의 디바이스에 하이브리드 시스템의 특성을 기반으로 하여 시뮬레이션 커널을 적절히 분배하는 과정이 매우 중요하며, 그 과정은 다음과 같다. 제안하는 시뮬레이션 구조에서 하이브리드 시스템 시뮬레이션의 가속화를 위하여, 시뮬레이션의 과정 중 이산 사건 시뮬레이션은 CPU에서 동작하도록, 연속 시간 시뮬레이션은 GPU에서 동작하도록 구현하였다. GPU는 특히 대규모 부동소수점 연산에 장점을 가지는 가속기로서 대규모 하이브리드 시스템에서 연속 시간 시스템의 시뮬레이션에 적합하다. 나누어진 시뮬레이션의 전체 프로세스는 Figure 3과 같이 멀티 코어와 GPU에 할당되어 동작하게 된다.

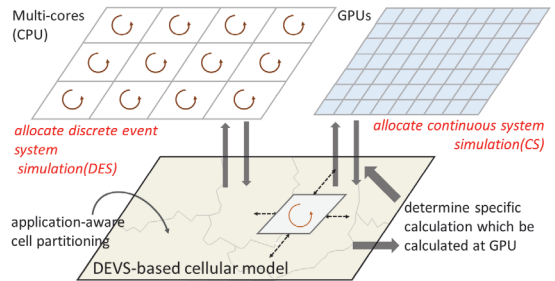


Fig. 3. Proposed CPU-GPU tightly coupled computation in DEVS based simulation

대규모 하이브리드 시스템을 멀티 코어와 GPU로 분배하는 구조를 셀룰러 모델을 그 예로 들어 설명하고자 한다. 본 연구에서 사용되는 셀룰러 모델은 다수의 셀 간의 연결로 구성되어 있고, 대규모 셀 모델의 각 셀은 연속 시간 시스템 모델링에 사용되는 미분 방정식에 의해 내부 변화 동작을 가진다. 따라서 각 셀의 연속 시간 시스템의 시뮬레이션 과정은 미분 방정식 계산을 따르며, 반복적이다. 이러한 이유로 연속 시간 시스템의 시뮬레이션 과정이 GPU에서 동작하게 된다면 그 결과는 시뮬레이션 시간의 관점에서 효과적인 결과를 얻게 됨을 예상할 수 있다. 다음은 연구에서 제안하는 DEVS 기반의 멀티 코어와 GPU 구조에서 고려해야할 부분이다. GPU는 장치로 계산할 데이터들을 복사해야만 한다. 즉, 이러한 이유로 GPU는 반복되는 대규모 연산에 장점을 가지는 반면, GPU로 데이터를 복사해가는 시간이 필요함을 적절히 고려해야 한다. 그러므로 만약 데이터 복사가 빈번하거나, 작은 양의 데이터를 복사한다면 GPU로 복사하는 과정은 전체 과정에 대해 비효율적일 수 있다. DEVS 기반의 하이브리드 시스템에서 GPU를 효과적으로 사용

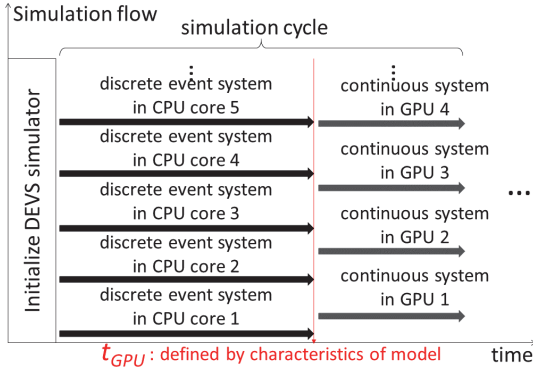


Fig. 4. Idea of using GPU in hybrid system simulation

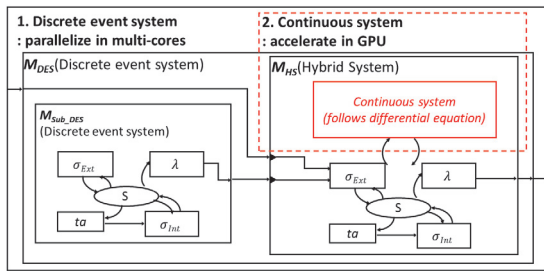


Fig. 5. Proposed architecture in terms of model specification

하기 위해, Figure 4와 같이, 시뮬레이션 대상인 시스템을 정확히 이해하고 그로부터 GPU에서 시뮬레이션 하고자 하는 동작의 시점을 정하는 과정이 필요하다. 대규모 셀룰러 모델의 시뮬레이션에서 본 연구에서 제안하는 GPU로의 데이터 복사 시점은 가장 일반적인 방법으로, 하이브리드 시스템의 이산 사건 시스템의 시뮬레이션이 끝난 후, 연속 시간 시스템 모델을 GPU에서 시뮬레이션 하는 방법이다. 이 방법을 적용하기 위해, 타겟 모델 특성의 정확한 이해와 더불어 DEVS 시뮬레이션의 흐름 제어가 필요하다.

본 연구에서 타겟 모델로 사용하는 셀룰러 모델은 많은 노드 사이에 다수의 연결이 존재하는 모델과 비슷하며, 일반적으로 미분 방정식의 계산이 포함된다. Figure 5는 연구에서 사용하는 모델의 기본적인 구조에 대하여 설명한다. 시뮬레이션 과정에서, 사용하는 대규모 셀룰러 모델은 Figure 5와 같이 수많은 하이브리드 시스템들의 연결로 구성되어 있다. 하이브리드 시스템의 시뮬레이션을 가속화하기 위하여 이산 사건 시스템의 시뮬레이션은 멀티 코어에서, 미분 방정식을 포함하는 연속 시간 시스템의 시뮬레이션은 GPU에서 동작하는 것을 제안한다. 또한, 멀티 코어에서 동작하는 이산 사건 시스템의 시뮬

레이션은 셀룰러 모델을 여러 하위 모델로 나누어 시뮬레이션 하도록 하였다. 셀룰러 모델은 그 지역적인 특징으로부터 여러 하위 모델로 나누어질 수 있으며, 각 하위 모델들을 코어에 할당하여 시뮬레이션하고, 각기 다른 코어에서 동작하는 하위 모델 간의 통신을 구현하여 전체 시뮬레이션의 동작을 가능하게 하였다.

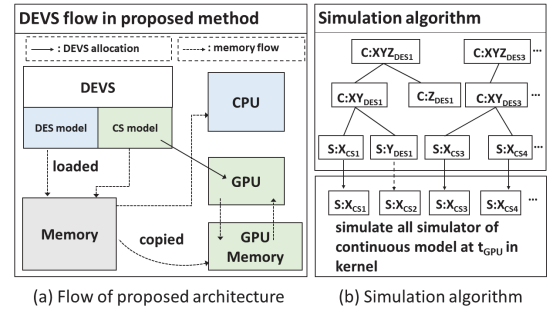


Fig. 6. Proposed architecture in terms of flow and algorithm

Figure 6 (a)는 이기종 아키텍처에서의 DEVS 동작 흐름을 나타낸다. DEVS 형식론 기반의 하이브리드 시스템은 시뮬레이션 동작 중에 메인 메모리에 로드된다. 이산 사건 시스템 모델은 CPU 코어에서 사용되기 위해 메모리에 할당 되고, 연속 시간 시스템 모델은 GPU의 메모리에 할당되어 사용된다. 이 과정에서 특히 연속 시간 시스템 동작을 위한 메모리는 GPU와 CPU 사이에서 복사되어지며 사용된다.

Figure 6 (b)는 제안하는 구조의 DEVS 시뮬레이터의 대략적인 알고리즘을 나타낸다. S:X는 원자 모델을 위한 시뮬레이터를 나타내며 C:XY는 결합 모델 XY와 대응되는 코디네이터를 나타낸다. 시뮬레이터는 원자 모델을 위하여 시뮬레이션 동작을 업데이트하고, 네 개의 원자 모델의 함수를 호출하여 상태를 천이한다. 코디네이터는 네 개의 결합 함수를 호출하며 메시지 전달을 담당한다. 연속 시간 시스템의 원자 모델을 위한 시뮬레이터는 GPU의 커널에서  $t_{GPU}$  시간에서 동작하며, 이 시간은 모든 이산 사건 시스템이 동작한 후의 시점이다. 시뮬레이터와 코디네이터에 의한 간략한 DEVS 시뮬레이션 알고리즘은 Figure 7의 알고리즘과 같다. 모델의 상태는 다가오는  $t_N$ 이 갖고 있는  $t_N$ 과 같아질 경우 변하게 된다. 즉, 제안하는 GPU의 동작 시점을 한 시점에서 모든 셀 모델들의 상태가 변하게 되는 마지막  $t_N$ 을 찾고, GPU 동작 시점에 적용하도록 한다. 즉, 이  $t_N$ 이 지난 후, 하이브리드 시스템에서 미분 방정식 형태를 가지는 연속 시간 시스템들

을 GPU CUDA의 쓰레드와 블록 인덱스를 이용한 GPU 동작을 수행하여 시뮬레이션 되도록 한다.

```

Algorithm 1 Proposed DEVS simulation cycle
for number of K cells
  for DES simulation
    coordinator.send( $t_N$ )
    simulator.reply( $next\ t_N$ )
    coordinator.compare( $next\ t_N$ )
    coordinator.send( $global\ t_N$ )
    simulator.check( $global\ t_N$ )
    if imminent
      simulator.computes( $output\_msg$ )
    end if
    simulator.send( $output\_msg$ )
    simulator.computes( $state$ )
  end for
  for CS simulation | by GPU kernel
    for  $i = 1$  to K,  $i$  is calculated by thread, block index
      simulator.compute( $CS\ model(i)$ )
    end for
  end for
end for
    
```

Fig. 7. Proposed simulation algorithm

## 5. 구현

### 5.1 멀티 코어와 GPU를 이용한 시뮬레이션

멀티 코어와 GPU를 이용한 제안하는 이기종 아키텍처의 시뮬레이션 동작 과정은 Figure 8 (a)와 같다. 이산 사건 시스템과 연속 시간 시스템이 결합된 하이브리드 시스템은 DEVS 엔진을 이용하여 시뮬레이션 되며, 미분 방정식을 포함하는 연속 시간 시스템은 GPU에서 시뮬레이션 된다. 이산 사건 시스템 모델을 이용한 시뮬레이션의 흐름은 하위 모델로 나눔으로써 멀티 코어에서 병렬 처리 되도록 하였다. 본 연구에서 시뮬레이션은 다수의 GPU 장치를 이용하여 동작하도록 하였고, 이산 사건 시

스템을 따르는 나머지 시뮬레이션 부분은 멀티 코어에 분배되어 동작하도록 하였다. 설정된 시간 간격이 흐른 후, 멀티 코어와 GPU에서 동작한 시뮬레이션 결과는 하나로 합쳐진다. 그 후 다음 시뮬레이션 동작이 수행된다. 전체 시뮬레이션 동작은 언급한 이 두 동작이 반복되며 진행된다. GPU 측의 처리 과정은 GPU CUDA를 이용하여 구현되었다.

### 5.2 대규모 셀룰러 모델 기반의 화재 확산 모델

제안하는 이기종 아키텍처를 분석하기 위하여 Figure 8 (b)와 같은 화재 확산 모델 현상을 시뮬레이션 하도록 한다. 이 시스템은 DEVS 형식론을 기반으로 모델링 되었으며, 각각의 셀들은 주변의 셀들로부터 임계 값 이상의 값을 전달 받을 경우 불안정한 상태로 변하게 된다. 이 상태 천이 과정은 연속 시간 시스템으로 모델링되며, GPU에서 제안하는 방법으로 시뮬레이션 된다.

각각의 셀들의 연속 시간 시스템의 미분 방정식은 Runge-Kutta 방법을 이용하여 구현되었다. 또한, 대규모 셀을 사용함으로써 보다 자세한 시뮬레이션 결과를 얻도록 하였다. 화재 확산 모델의 시뮬레이션 과정은 크게 두 부분, 주변 셀과 메시지 전달을 포함한 전체 동작 부분과 내부, 외부 상태에 의해 상태가 변하는 부분으로 나뉜다. 이 두 부분은 시뮬레이션의 가속화 지점에서 각각의 특성에 따라 멀티 코어와 GPU로 각각 분배되어 처리된다. DEVS 형식론 기반의 하이브리드 시스템을 이용한 화재 확산 모델은 Figure 9와 같이 나타난다.  $M_{cell}$ 은  $M_{gen}$ 으로 구성되며, 내부 상태 변화를 위한 상태를 생성한다.  $M_{HS}$ 는 그 자신의 상태 변화 및 주변의 셀들로부터 상태를 받아 변화한다. 모델들 중에서,  $M_{HS}$ 는 하이브리드 시스템 명세를 통해 모델링 되었다. 하이브리드 시스템  $M_{HS}$ 의

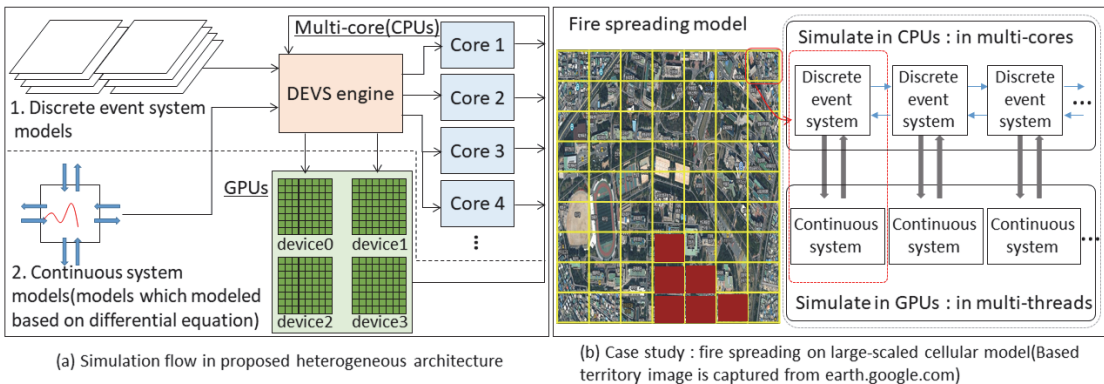


Fig. 8. Proposed simulation flow and case study

수학적 표현은 아래와 같다.

- $M = \langle X, Y, S_{DES}, \delta_{ext}, \delta_{int}, \lambda, ta, S_{CS}, F, g, \phi \rangle$
- $X$  : 이산 사건 입력 집합
- $Y$  : 이산 사건 출력 집합
- $S_{DES}$  : 이산 사건 상태의 집합
- $\delta_{ext} : Q_{DES} \times Y_{CS} \rightarrow S_{DES}$  : 외부 상태 천이 함수
- $\delta_{int} : Q_{DES} \rightarrow Q_{DES}$  : 내부 상태 천이 함수
- $\lambda : Q_{DES} \rightarrow Y$  : 출력 함수
- $ta : S_{DES} \rightarrow R_{0, \infty}^+$  : 시간 진행 함수
- $S_{CS}$  : 연속 상태의 집합
- $F$  : 임계값의 집합
- $g : Q_{CS} \rightarrow Y_{CS}$  : 연속 시간 시스템의 출력 함수
- $\phi$  : 미분 방정식의 집합

$X$ 는 주변 셀들로부터 전달되는 입력의 집합을 나타내며,  $Y$ 는 출력 집합을 나타낸다.  $S_{DES}$ 는 이산 사건 시스템의 상태, wait, done, stable unstable을 나타낸다. 이 상태 값들은  $\delta_{ext}$ 와  $\delta_{int}$ 에 의해 업데이트 되고, 현재 셀의 상태의 안정성, 그리고 동작의 동기화에 사용된다.  $\lambda$ 는 출력 함수이며,  $ta$ 는 이산 사건 시스템의 시간 진행 함수이다. 연속 시간 시스템에 대하여  $S_{CS}$ 는 마찬가지로 시스템의 상태의 집합이며,  $F$ 는 임계값의 집합으로, 연속 상태에서 이벤트로의 변환에 이용된다. 동시에  $g$ 는 출력 함수이며,  $\phi$ 는 연속 시간 시스템의 미분 방정식을 나타낸다.

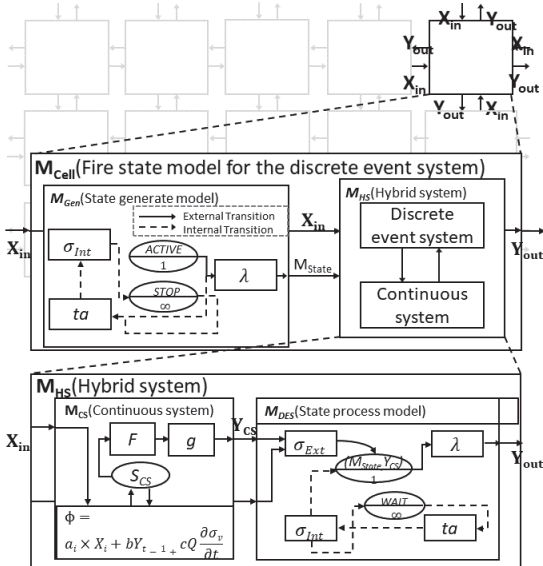


Fig. 9. DEVS based fire spreading model of large-scaled hybrid system

## 6. 실험 결과

본 연구에서는 제안하는 구조를 Table 1에 나타난 하드웨어들의 여러 조합에 대한 시뮬레이션 시간과 디바이스들의 최대 전력 소모 관점에서 분석하였다. Figure 10은 제안하는 이기종 아키텍처 기반의 시뮬레이션에서 사용된 장치들을 나타낸다. 제안하는 구조에서의 시뮬레이션을 Table 2와 같이 8가지 경우의 하드웨어 조합에 대해 나누고, 각각 동작하였다. 시뮬레이션의 실행 시간은 셀의 개수, 하드웨어 조합, Runge-Kutta 방법을 이용 미분 방정식에서의 시간 간격을 바꾸어가며 측정되었다. 추가로, 8가지 하드웨어 조합과 여러 실험 조건에 대하여 최대 전력 소모를 각각 분석하고자 하였다. 본 연구의 실험 부에서 측정된 시뮬레이션 시간은 모든 셀들이 전부 한번씩 갱신되는 시간을 측정하였다.

Table 1. Hardware specifications

Devices	Model	Cores
CPU	Intel Xeon CPU E5-2620 v4 × 2	16
GPU	NVIDIA GeForce GTX 1080 Ti × 4	14336
Memory	16GB × 8	-

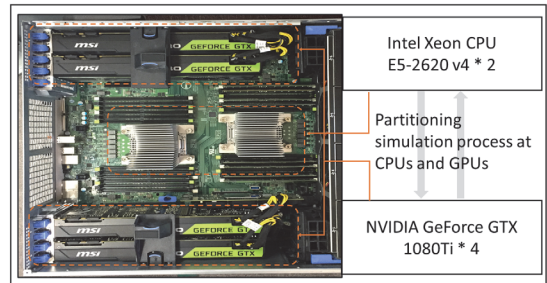


Fig. 10. Target hardware for computing proposed simulation framework

Table 2. Hardware configurations

Case	Devices
1	4 CPU cores
2	8 CPU cores
3	12 CPU cores
4	16 CPU cores
5	4 CPU cores, 3584 GPU cores
6	8 CPU cores, 7186 GPU cores
7	12 CPU cores, 10752 GPU cores
8	16 CPU cores, 14336 GPU cores

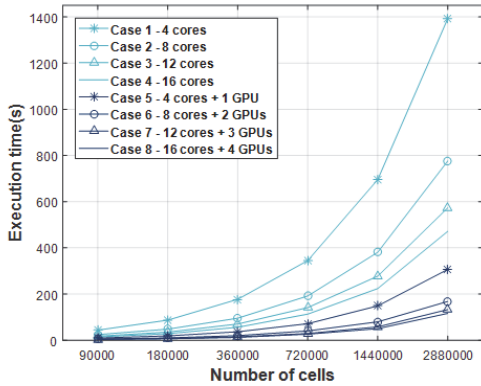


Fig. 11. Execution time of simulation according to the number of cells

Figure 11은 셀의 개수와 하드웨어 조합을 바꾸어가며 측정한 시뮬레이션의 시간이다. 8가지 경우의 하드웨어 조합에 대하여 시뮬레이션 시간을 측정하였으며, 셀의 개수를 90000개에서 2880000개로 변경해가며 시간을 측정하였다. 미분 방정식 계산을 위한 연속 시간 시스템의 시간 간격은 0.0001을 사용하였다. 그 결과, 셀의 개수가 늘어날수록 시뮬레이션 시간은 더 가파르게 증가하는 것을 확인할 수 있다. 같은 CPU 코어 개수를 사용하는 4번과 8번의 하드웨어 조합의 경우, 8번의 하드웨어 조합의 시뮬레이션 시간이 180000개의 셀을 사용하였을 때 10% 빠른 실행 시간을 보였고, 2880000개의 셀을 사용하였을 경우 80% 더 빠른 시간에 동작함을 확인하였다. 즉, CPU-GPU가 결합된 제안하는 구조의 시뮬레이션이 대규모 셀의 시뮬레이션에 이점을 가지는 것을 확인하였다.

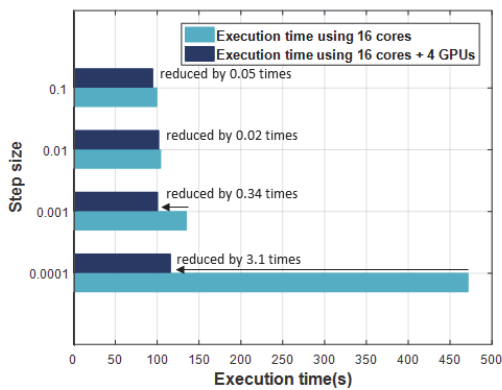


Fig. 12. Execution time of simulation according to the step size of differential equations

시간 간격을 기반으로 연산되는 Runge-Kutta 방식의 정확도는 시간 간격의 크기에 따라 정확도가 달라지는 것으로 알려져 있다. 일반적으로 더 조밀한 시간 간격을 이용한 연산이 더 정확한 결과를 얻어낸다. 실험에서는 1차 미분 방정식 기반의 연속 시간 시스템을 구현하였으며, 시뮬레이션 시간은 Figure 12와 같이 시간 간격을 바꾸어가며 측정되었다. 16개의 CPU 코어와 4개의 GPU를 이용하여 실행 시간을 측정하였으며, 미분 방정식 계산을 위한 시간 간격은 0.1에서 0.0001로 변경해가며 측정하였다. 그 결과 제안하는 CPU-GPU가 결합된 시뮬레이션과 CPU만 사용한 시뮬레이션의 시간과 비교 하였을 때, 제안하는 구조의 시뮬레이션 시간이 미분 방정식 계산의 시간 간격이 조밀해짐에 따라 0.05배에서 3.1배 만큼 상대적으로 적음을 확인하였다. 또한, CPU-GPU가 결합된 시뮬레이션의 실행 시간은 시간 간격이 변하더라도 거의 일정한 것을 확인하였다. 16개의 CPU 코어만을 이용하는 4의 경우 시간 간격에 대한 시뮬레이션 시간의 차이가 확연하게 드러남을 확인하였고, 0.1 시간 간격을 이용한 시뮬레이션과 0.0001 시간 간격을 이용한 시뮬레이션의 시간은 약 3.7배가 차이 나는 것을 확인하였다. 반면, 제안하는 CPU-GPU 구조에서의 실행 시간은 같은 조건에 대하여 0.02배 차이가 나타났으며, 그 결과 제안하는 구조에서 GPU가 대규모 부동소수점 연산에 장점을 가지며 연속 시간 시스템의 시뮬레이션에 적합한 것을 확인하였고, 연속 시간 시스템의 시뮬레이션 시간 관점에서 매우 효율적인 구조임을 확인하였다. 추가로 제안하는 구조의 GPU에서 사용하는 추가 메모리 사용량은 Table 3과 같으며, 각 GPU 장치는 680 MBytes의 추가 메모리 사용량을 가진다. 즉, 8번 경우, 약 2.5 GBytes의 추가 메모리 사용량을 가지는 것을 알 수 있다. GPU에서 사용되는 추가 메모리 사용량으로 인해 시뮬레이션 과정에 사용할 수 있는 시스템의 메모리 사용량이 제한될 수 있지만 그 양이 크지 않은 것을 확인하였다. 즉, 제안하는 이기종 아키텍처를 이용한 시뮬레이션의 경우 추가 메모리 사용량을 가지지만 실행 시간 관점에서 성능이 크게 향상되는 것을 확인하였다.

Table 3. Additional memory usage

Hardware configurations	Memory usage in GPU
4 cores + 1 GPU	686 MBytes
8 cores + 2 GPUs	1308 MBytes
12 cores + 3 GPUs	1938 MBytes
16 cores + 4 GPUs	2552 MBytes



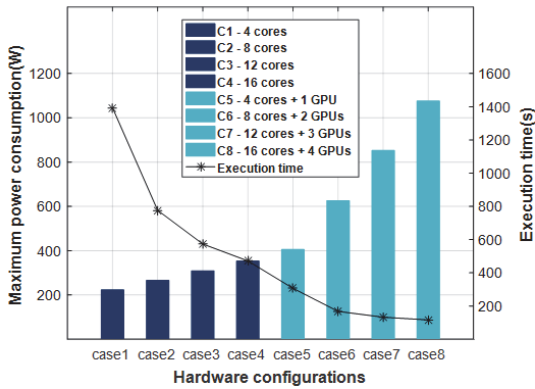


Fig. 13. Maximum power consumption and execution time according to the hardware configurations

제안하는 시뮬레이션의 전력 관점에서의 효율성을 분석하기 위해 Figure 13과 같이 8가지 하드웨어 조합에 대하여 최대 전력 소모를 측정하고 분석하였다. CPU 코어를 이용한 시뮬레이션의 경우 전력 소모는 이론적으로 구하였으며, GPU를 사용한 시뮬레이션의 경우 nvidia-smi 툴을 이용하여 실제로 구하였다. 실험은 2880000개의 셀과 0.0001 시간 간격, 그리고 8개의 하드웨어 조합을 변경해가며 진행하였다. 최대 전력 소모와 시뮬레이션 실행 시간은 그래프와 같이 반비례하는 것을 확인할 수 있으며, 더 많은 전력이 소모될 경우 실행 시간 관점에서 더 높은 성능을 가지는 것을 확인하였다. 본 연구에서 측정 한 전력 소모량은 최대 전력 소모 값이며 시뮬레이션에 소모되는 전체 전력의 경우, 제안하는 구조에서의 전력 소모량이 시뮬레이션 시간이 길어질수록 상대적으로 더 작아져 전체 전력 소모 관점에서도 이점을 가지는 것을 알 수 있다. 하드웨어 조합 4번의 경우와 5번의 경우에 대해, 5번의 경우 405 와트를 소모하였으며, 4번의 경우 353 와트를 소모하였다. 즉, 5번 조합의 경우 4번 보다 1.1배의 전력을 더 소모하였지만, 반면 시뮬레이션 시간은 1.5배로 감소하였다. 실행 시간과 전력 소모 차이가 가장 큰 1번 경우와 8번 경우, 8번의 경우 약 3.8배의 전력 소모를 가지지만, 시뮬레이션 시간은 약 11배 빨라지는 것을 확인할 수 있다.

## 7. 결론

본 논문에서는 DEVS 형식론 기반의 하이브리드 시스템의 가속화를 위해 CPU와 GPU가 결합된 이기종 아키텍처에서의 시뮬레이션을 제안하였다. 타겟 시스템 모델

로 대규모 셀룰러 모델을 이용하였으며, 그 중 화재 확산 모델을 예로 사용하였다. 전체 시뮬레이션 과정은 하이브리드 시스템의 특성에 따라 연속 시간 시스템과 이산 사건 시스템으로 분할하였다. 대규모 셀룰러 모델에서 동일한 연산이 반복되는 특성을 가지는 연속 시간 시스템의 시뮬레이션은 GPU에 할당하여 동작하도록 하였으며, 이산 사건 시스템은 CPU의 멀티 코어에서 병렬 처리 하였다. 제안하는 구조의 시뮬레이션을 여러 하드웨어 조합과 셀의 개수를 변경해가며 분석하였으며, 전력 소모 관점에서 분석하여 효율적인 시뮬레이션의 하드웨어 조합을 찾고자 하였다. 또한, DEVS 형식론 기반의 하이브리드 시스템에서 연속 시간 시스템의 시뮬레이션 정확도를 위한 미분 방정식의 시간 간격을 변경해가며 시뮬레이션 시간을 분석하였다. 그 결과, 제안하는 구조가 DEVS 기반의 가속화된 하이브리드 시스템의 시뮬레이션에 적용 가능함을 확인하였고, 제안하는 이기종 아키텍처를 실행 시간 대 전력 소모 관점에서 분석하여 효율적인 시뮬레이션 구조임을 확인하였다. 추후 연구로는 제안하는 CPU-GPU 결합 시뮬레이션 구조를 다수의 레이어를 포함하는 시뮬레이션에 적용하고 넓혀나갈 계획이다.

## References

- B. P. Zeigler, H. S. Song, T. G. Kim, and H. Praehofer. (1995), "Devs framework for modelling, simulation, analysis, and design of hybrid systems", in *Hybrid Systems II*, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 529-551.
- F. Bergero and E. Kofman. (2011), "Powerdevs: a tool for hybrid system modeling and real-time simulation", *SIMULATION*, vol. 87, no. 1-2, pp. 113-132.
- J. H. Hong, K.-M. Seo, and T. G. Kim. (2013), "Simulation-based optimization for design parameter exploration in hybrid system: a defense system example", *SIMULATION*, vol. 89, no. 3, pp. 362-380.
- G. A. Wainer. (2004), "Modeling and simulation of complex systems with cell-devs", in *Proceedings of the 2004 Winter Simulation Conference*, 2004, vol. 1, Dec, p. 60.
- Q. Liu and G. Wainer. (2007), "Parallel environment for devs and cell-devs models", *SIMULATION*, vol. 83, no. 6, pp. 449-471.

- Q. Liu and G. Wainer. (2012), "Multicore acceleration of discrete event system specification systems", *SIMULATION*, vol. 88, no. 7, Jul. 2012, pp. 801-831.
- M. G. Seok and T. G. Kim. (2012), "Parallel discrete event simulation for devs cellular models using a gpu", in *Proceedings of the 2012 Symposium on High Performance Computing, ser. HPC '12. San Diego, CA, USA: Society for Computer Simulation International*, pp. 11:1-11:7.
- H. Saadawi and G. Wainer. (2012), "On the verification of hybrid devs models", in *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium, ser. TMS/DEVS '12. San Diego, CA, USA: Society for Computer Simulation International*, pp. 26:1-26:8.
- A. C. Chow, B. P. Zeigler and D. H. Kim. (1994), "Abstract simulator for the parallel devs formalism", in *Fifth Annual Conference on AI, and Planning in High Autonomy Systems*, Dec, pp. 157-163.
- S. Jafer and G. Wainer. (1980), "Flattened conservative parallel simulator for devs and cell-devs", in *2009 International Conference on Computational Science and Engineering*, vol. 1, Aug, pp. 443-448.
- J. Himmelspach, R. Ewald, S. Leye, and A. M. Uhrmacher. (2007), "Parallel and distributed simulation of parallel devs models", in *Proceedings of the 2007 Spring Simulation Multiconference - Volume 2, ser. SpringSim '07. San Diego, CA, USA: Society for Computer Simulation International*, pp. 249-256.
- A. Troccoli and G. Wainer. (2003), "Implementing parallel cell-devs", in *Proceedings of the 36th Annual Symposium on Simulation, ser. ANSS '03. Washington, DC, USA: IEEE Computer Society*, pp. 273
- A. C. H. Chow and B. P. Zeigler. (1994), "Parallel devs: a parallel, hierarchical, modular modeling formalism", in *Proceedings of Winter Simulation Conference*, Dec, pp. 716-722.
- H. Park and P. A. Fishwick. (2010), "A gpu-based application framework supporting fast discrete-event simulation", *SIMULATION*, vol. 86, no. 10, Oct. pp. 613-628.
- G. A. Wainer and N. Giambiasi. (2001), "Application of the cell-devs paradigm for cell spaces modelling and simulation", *SIMULATION*, vol. 76, no. 1, pp. 22-39.
- G. A. Wainer and N. Giambiasi. (2002), "N-dimensional cell-devs models", *Discrete Event Dynamic Systems*, vol. 12, no. 2, Apr, pp. 135-157.
- L. Ntaimo, B. P. Zeigler, M. J. Vasconcelos, and B. Khargharia. (2004), "Forest fire spread and suppression in devs", *SIMULATION*, vol. 80, no. 10, pp. 479-500.
- A. Muzy, E. Innocenti, A. Aiello, J.-F. Santucci, and G. Wainer. (2002), "Methods for special applications: Cell-devs quantization techniques in a fire spreading application", in *Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers, ser. WSC '02. Winter Simulation Conference*, pp. 542-549.
- X. Hu and B. P. Zeigler. (2004), "A high performance simulation engine for large-scale cellular devs models", in *Proceedings of the 2004 Symposium on High Performance Computing*, pp. 3-8.



**김 성 섭** (kss92318@gmail.com)

2017 경북대학교 전자공학부 학사  
2017~ 현재 경북대학교 전자공학부 석사과정

관심분야 : 고성능 임베디드 컴퓨팅, 시뮬레이션, 하드웨어-소프트웨어 Co-design, Dependable 스마트 IoT 시스템



**조 정 훈** (jcho@ee.knu.ac.kr)

1996 한국과학기술원(KAIST) 전기 및 전자공학과 학사  
1998 한국과학기술원(KAIST) 전기 및 전자공학과 석사  
2003 한국과학기술원(KAIST) 전기 및 전자공학과 박사  
2003~ 2005 SK Hynix 선임연구원  
2005~ 현재 경북대학교 전자공학부 교수

관심분야 : 임베디드 시스템, 바이너리 변환, 차량용 안전 및 보안시스템, AUTOSAR, 런타임 감시



**박 대 진** (boltanut@knu.ac.kr)

2001 경북대학교 전자전기공학부 학사  
2003 한국과학기술원(KAIST) 전기 및 전자공학과 석사  
2003~ 2004 한국과학기술원(KAIST) 전기 및 전자공학과 박사  
2003~ 2014 SK Hynix/Samsung (차세대 LSI 설계) 수석연구원  
2014~ 2016 경북대학교 전자공학부 초빙교수 (2014년 대통령 Postdoctoral Fellow 선정)  
2016~ 현재 경북대학교 전자공학부 조교수

관심분야 : 저전력 SoC 설계, 하드웨어-소프트웨어 Co-design, Dependable 스마트 IoT 시스템, Robust 임베디드 시스템