

Attributed AND-OR Graph : 디지털 시스템 설계에 있어 모델 관리를 위한 정형론

김준경*, 김탁곤*

Attributed AND-OR Graph : A Semantics for Formal Model Management for Digital Systems Design

Jun Kyoung Kim, Tag Gon Kim

Abstract

The progress of silicon technology enables to implement a highly complex digital system on a given chip area. However, even the modern design environment is not so efficient to catch up with the progress of process technology. Design reuse is a promising approach to designing such a complex system in an efficient way. However, the rigidity and inflexibility of a model has been an obstacle to design reuse.

This paper proposes a high-level model management methodology by introducing attributed AND-OR graph(AOG), a formal semantics for representing the possible structure of a model. Using the formalism enables a designer to extract, extend and reuse the pre-modeled and pre-verified design. A complete process of constructing a cache operational model, extending the model and extracting executable models is exemplified to show effectiveness of the proposed framework.

Key Words: Attributed AND-OR Graph, Model Management, Design Reuse, Digital System Design Methodology, Cache Model, SES, Intellectual Property

* 한국과학기술원

1. 서론

공정 기술의 발달로 인하여 주어진 칩 공간에 보다 많은 기능을 넣는 것이 가능해졌다. 1980년대에 1~3 micron 공정하에서 수천 개의 트랜지스터를 설계할 수 있었던 것에 비해 2000년 들어서는 수백million의 transistor를 집적할 수 있게 되었다. 이러한 추세로 인하여 현재 하드웨어를 설계하는 것은 기존의 설계 기술로는 감당하기 어려울 정도로 매우 어렵다. 또한 더 중요해지고 있는 time-to-market을 만족시키기도 어려워지고 있다.

설계해야 하는 시스템이 점점 더 복잡해지고 있다는 것은 시스템 설계 도구에 설계 시간을 단축하여야 한다는 challenge를 주고 있다. 설계 시간을 줄이기 위한 한 가지 방법으로, 보다 높은 추상레벨에서 시스템을 설계하는 방법이 꾸준히 연구되어 왔다.

설계 시간을 줄이기 위한 다른 방법으로는 IP(Intellectual Property) 재사용이 있다. 이는 새로운 시스템을 설계할 때 기존에 이미 설계되고 검증된 모델을 그대로 재사용함으로써 전체 설계 시간을 줄이는 데 도움이 된다. [2]는 IP를 사용하는 데에 있어서 제한된 option, 테스트의 어려움, 구현과 모델의 불일치 및 copyright과 같은 많은 어려움이 있을 것이라고 예상한다. 이러한 어려움에도 불구하고 [2]는 다음과 같이 결론을 내리고 있다. *"going forward, the decision to use a synthesizable core will continue to provide benefits"*.

2. 모델 재사용

모델 재사용성의 중요성은 예전부터 강조되어 왔다. 이를 위해서 상당히 많은 연구기관이 노력하고 있다. 국내 기관으로는 SIPAC(System Integration & Intellectual Property Authoring Center)와 같은 기관이 있고 그 외에 미국의 VSIA(Virtual Socket Interface Alliance)나 OCP-IP(Open Core Protocol International Partnership)와 같은 기관에서 모델 재사용을 위해 표준화 작업을 수행하고

있다.

그러나 기존에 설계된 모델을 재사용한다는 것은 개념처럼 깨끗하게 이루어지고 있지는 않는 것이 현실이다. 우선 기술적으로는 interface, 혹은 모델의 보안과 같은 문제를 해결해야 하고 법적으로는 copyright와 같은 여러 가지 문제가 선결되어야 한다. 또한 많은 경우에 있어 새로운 시스템을 설계하기 위해 기존에 설계된 모델을 재사용하는 과정에서 기존의 모델을 조금도 고치지 않고 그대로 사용하는 일은 극히 드물다. 설계자의 목적에 따라, 혹은 성능을 향상시키기 위해, 혹은 새로운 표준을 만족시키기 위해 기존의 모델을 고치고 재구축하는 일이 빈번히 일어난다. 그러나 대개의 IP를 제공하는 측에서는 단순히 어떤 표준화된 틀에 맞추어 구현된 IP들을 특정 category로 나누어 저장하고, 있는 그대로 사용자에게 제공하고 있다.

본 연구에서는 모델의 재사용성을 높이기 위해 모델을 상위 레벨에서 관리하는 기법을 제안한다. 우선 모델의 구조를 두 개의 관계, 즉 and-relation과 or-relation을 사용하여 계층적으로 나타낸다. 하나의 모델을 가능한 한 독립적인 부분으로 나누고 각 부분에 대해서도 구현 가능한 다양한 방법을 모두 기술함으로써 하나의 모델의 전체 구조를 기술하게 된다. 이를 통해서 크게 두 가지의 장점을 얻을 수 있다. 첫 번째로는 모델 구조를 계층적으로 표현하기 때문에 전체 구조의 어느 부분에서부터라도 재사용이 가능해진다. 그 결과로 재사용성의 향상을 얻을 수 있다. 두 번째로는 모델을 쉽게 확장할 수 있다는 장점이 있다. 전체 모델을 독립적인 부분으로 나누기 때문에 전체 모델의 일부분을 고쳐야 하는 필

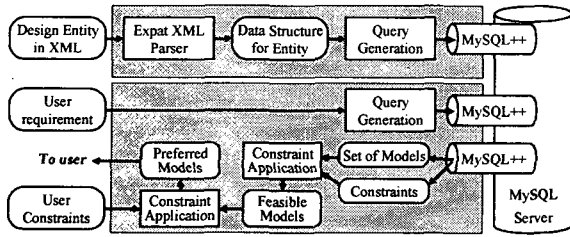


그림 1 제안하는 환경의 전체 구조요성이 생기는 경우 해당 부분에 집중할 수 있다.

그림 1은 본 논문에서 제안하는 환경의 전체 구조를 나타내고 있다. 모델의 구조를 위한 semantics로는 attributed AND-OR graph를, syntax는 XML (eXtensible Meta Language)를 사용하여 나타낸다[3]. 이 모델을 XML 파서를 이용하여 데이터 구조로 변환하고 back-end의 database에 저장한다. 이를 위해서 MySQL database management system를 사용하였고 MySQL과 통신하기 위해 MySQL++ library를 사용하였다. 구축된 model database로부터 원하는 모델을 추출하기 위하여 사용자는 자신의 requirement를 기술하고 이를 사용하여 사용자가 원하는 전체 model을 가져오게 된다. 하나의 모델은 보다 작은 모델의 구성으로 이루어지기 때문에 다양한 형태의 constraint가 존재할 수 있다. 이를 적용시켜 feasible model들을 얻게 된다. 이에 사용자가 자신만의 constraint를 작성하고 적용시켜 최종적인 preferred model을 추출하게 되고 이를 사용하게 된다.

3. High-level Model Management

3.1 Attributed AND-OR Graph

Attributed AND-OR graph는 모델의 구조를 나타내기 위해 도입한 정형론으로 모델의 전체 구조를 두 가지 관계를 이용하여 나타낸다. 하나는 AND-relation으로써 parent는 children의 조합으로 구성된다는 것을 의미한다. 다른 표현으로는 “decomposition-relation”이 있다. OR-rule은 parent를 구현하기 위해서 children중 하나를 선택하면 된다는 것을 의미한다. 이 관계는 “specialization-relation”

으로 표현할 수 있다. Attributed AND-OR graph는 다음과 같이 수학적으로 정의된다.

Attributed AND-OR Graph is a graph $G = (V, E_{AND}, E_{OR})$ where

* V : a vertex set

- 각 vertex는 전체 구조를 구성하는 entity이다.

- 각각의 vertex는 특성(attribute)를 가질 수 있고 다음과 같이 정의된다.

$$v = NAME_SET \times \prod_{i=1}^{num_attr} param_i$$

- param은 각 entity에 지정될 수 있는 파라미터이고 다음과 같이 정의한다.

$$param = NAME_SET \times POSSIBLE_SET$$

* $E_{AND} \in V \times I_1^\infty \times V$ where I_1^∞ is an integer more than zero: AND-relation

- E_{AND} 에 속하는 (v, i, v') 는 v 를 구성하기 위하여 i 개의 v' 가 aggregate되어야 한다는 것을 의미한다.

* $E_{OR} \in V \times V$: OR-relation

- E_{OR} 에 속하는 (v, v') 는 v' 는 v 를 구현하기 위한 하나의 선택이 될 수 있다는 것을 의미한다.

그림 2(a)는 디지털 시스템의 일반적인 구조이고 그림 2(b)는 이를 attributed AND-OR graph로 나타내고 있다. 는 parent와 children사이의 관계가 OR-relation이라는 것을 의미하고 는 parent와 children사이의 관계가 AND-relation이라는 것을 나타낸다. 가령 그림 2(a)에서 total_system entity는 proc, peripheral 및 memory_subsystem이라는 3가지의 entity로 구성된다는 것을 볼 수 있다. Proc entity에 대해서는 2 가지 종류의 core중 하나의 방법으로 구현할 수 있다는 것을 의미한다. 그림 2(b)의 total_system node정보는 아래와 같이 기술된다.

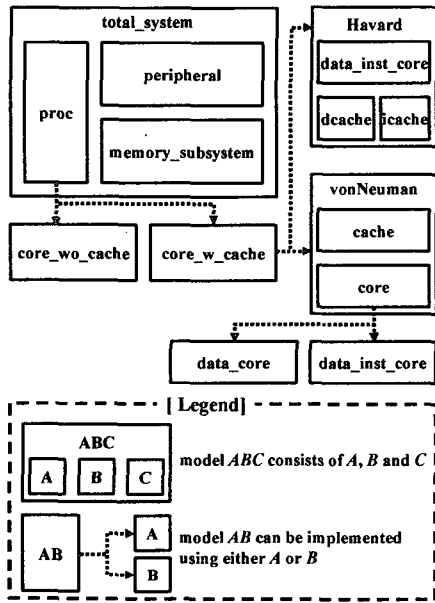
$(total_system, 1, proc), (total_system, 1, peripheral), (total_system, 1, memory_subsystem)$

Vertex에는 지정된 Attribute를 사용하여 디지털 시스템을 설계할 때 고려해야 하는 파라미터를 명시할 수 있다. 예를 들면 그림 2(b)

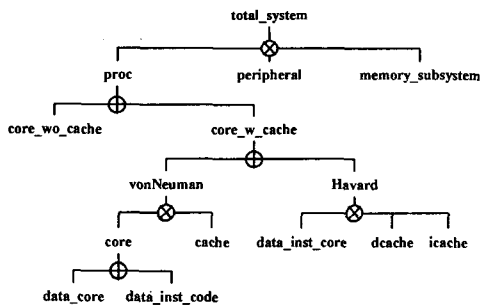
의 proc에는 *word_size*라는 attribute가 지정되어 있다. 만약 *word_size*라는 파라미터가 16, 32, 64과 같은 값을 갖는다면 정의에 따라 다음과 같은 attribute가 proc 노드에 기술된다.

{(*word_size*, 16), (*word_size*, 32), (*word_size*, 64) }

이런 방법으로 하나의 모델을 다양한 구조와 파라미터를 표현할 수 있도록 구축할 수 있다. 사용자는 or-relation의 children중 자신의



(a) 디지털 시스템의 구조



(b) (a)를 표현하는 attributed AND-OR graph

그림2 예제: 디지털 시스템과 AOG표현

목적에 맞는 노드를 선택하고 파라미터를 선택함으로써 원하는 모델을 얻게 된다.

3.2 Constraints on Models

하나의 vertex, 혹은 vertex간의 관계에 대한 constraint를 지정할 수 있다. Constraint의

정의는 아래와 같다.

const is defined for the vertex/vertices of attributed AND-OR Graph $G = (V, E_{AND}, E_{OR})$ as

- $const : V^n \rightarrow \{true, false, do_not_care\}$ where n is a non-negative integer

- $const(v_1, \dots, v_n) = true$: v_1, \dots, v_n 모두 있는 constraint는 항상 선택한다.

- $const(v_1, \dots, v_n) = false$: v_1, \dots, v_n 모두 있는 constraint는 절대로 선택하지 않는다.

- $const(v_1, \dots, v_n) = do_not_care$: 위와 같은 관계가 없음을 표시한다.

constraint로 vertex간에 물리적, 혹은 논리적으로 valid한 configuration을 정의할 수 있다. 예를 들면 shelving scheme을 사용하지 않는 superscalar 프로세서의 경우 항상 in order issue를 사용한다. 이 때 다음과 같은 constraint들 중 하나를 사용함으로써 생성 단계에서 invalid한 configuration을 배제할 수 있다.

$const(no_shelving, out_of_order_issue) = false$
 $const(no_shelving, in_order_issue) = true.$

3.3 SES와의 비교

본 논문에서 제안한 attributed AND-OR

SES	AOG
decomposition-relation	and-relation
multiple-decomposition	
specialization-relation	or-relation
attribute	attribute

표 1 SES와 AOG의 비교

graph는 SES(System Entity Structure)와 많은 면에서 유사하다[4]. SES는 어떤 system, 구조를 decomposition-relation, specialization-relation, 그리고 multiple-decomposition으로 표현한다. 본 논문에서 제안하는 attributed AND-OR graph는 이 SES를 기반으로 시스템 설계에 보다 적합한 형태로 확장한 것이다. SES와 attributed AND-OR graph의 대응관계는 표 1과 같다.

또한 SES의 특징인 uniformity, strict

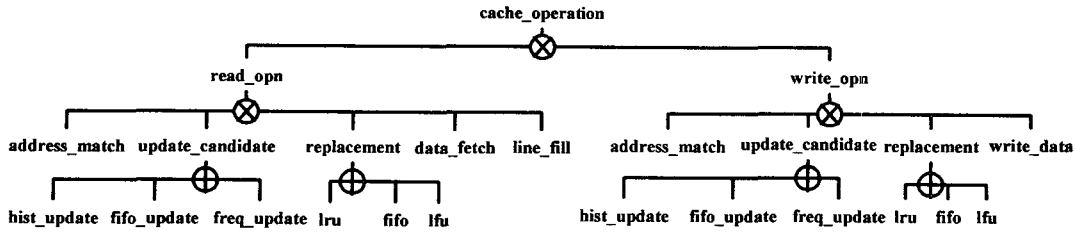


그림 3 Cache-operation 모델

hierarchy, alternating mode, valid brothers, attached variable property는 AOG 역시 만족하여야 한다. AOG를 SES로부터 구분할 수 있는 특징은 아래와 같다.

- and-relation을 확장하여 multiple-decomposition을 수용하였다. 즉 $(v,i,v') \in E_{AND}$ 에서 i 가 1인 경우 SES의 decomposition-relation, 1보다 큰 경우 multiple-decomposition에 해당한다.
- SES의 attribute에 해당하는 파라미터 역시 가능한 값들을 표현할 수 있게 함으로써 디지털 시스템 설계에 적합하게 정의하였다.
- $(v,i,v') \in E_{AND}$ 에서 number of copy인 i 를 attribute인 파라미터와 연계하여 명시할 수 있다. 즉 attribute를 이용하여 시스템의 전체 구조를 parameterize할 수 있다.

4. 예제 : Cache 모델 관리

본 연구에서 디지털 시스템 모델을 상위 수준에서 관리하는 예를 보이기 위해 cache model을 예제로 하였다. 그림 3은 구축된 cache operation model을 attributed AND-OR graph로 표현한 것이다. 이 모델이 의미하는 것은 cache operation은 read/write operation으로 이루어지고, cache read operation은 address_match, update_candidate, data_fetch, line_fill 및 replacement라는 보다 작은 모델들로 이루어져 있다는 것을 의미한다. update_candidate를 위해서는 3개의 구체적인 모델, 즉 hist_update, fifo_update 및 freq_update중 하나를 이용하여 구현되어야 한다. constraint specification을 이용하여 update_candidate의 children들과 replacement

의 children사이의 관계를 명시하였다. 또한 cache의 다양한 파라미터들은 cache_operation 노드의 attribute로 표현된다. 본 예제를 구축할 때에는 다음과 같은 파라미터를 명시하였다.

- cache_size : 4/8/16/32 K-bytes
- line_size : 4/8/16/32 bytes
- associativity : direct-mapped, 2/4/8/16-way set-associative, fully-associative

4.1 모델의 확장

앞서 설명한 바와 같이 많은 경우에 있어 기존의 모델에 새로운 방법이나 알고리즘을 추가하는 경우가 자주 발생한다. 본 연구에서 제안한 방법은 이러한 경우에 효과적으로 대응할 수 있다. 가령 그림 3에서 보이고 있는 cache operation 모델에 새로운 replacement 알고리즘을 추가하는 상황을 가정하자. 이 경우 추가되는 알고리즘에 의해 영향을 받는 것은 update_candidate, replacement 노드이다. 이러

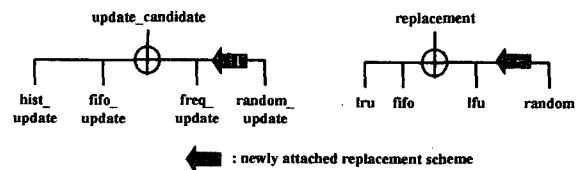


그림 4 모델의 확장

한 확장을 위하여 설계자는 다음과 같은 작업을 수행하면 된다.

- random_update, random 에 해당하는 모델을 작성한다.
- 새로이 작성된 모델은 적당한 위치에 or-relation으로 반영한다.

그림 4는 이러한 일련의 과정을 attributed

AND-OR graph 형태로 보이고 있다. 실제 이 전체 과정은 10분 이내로 수행되었으며 모델을 추출하여 실험해 본 결과 새로운 알고리즘이 완벽히 동작하는 것을 확인할 수 있었다.

4.2 모델의 추출

위와 같이 구축된 모델 database로부터 사용자는 원하는 모델을 추출하게 된다. 그림 5(a)는 이 과정을 자세히 보이고 있다. Phase 1에서는 database로부터 모든 설계를 attributed AND-OR graph 형태로 재구축한다. Phase 2에서는 attributed AND-OR graph가 의미하는 모든 설계 후보들을 생성한다. 하나의 설계 후보를 골라서 설계 tree를 구축한 후 파라미터를 하나로 선택하게 된다. 이렇게 선택된 모델의 구조와 파라미터를 가지고 SystemC 모델을 재구성 및 컴파일을 함으로써 실행 가능한 모델을 얻게 된다. 실제로 이 일련의 과정은

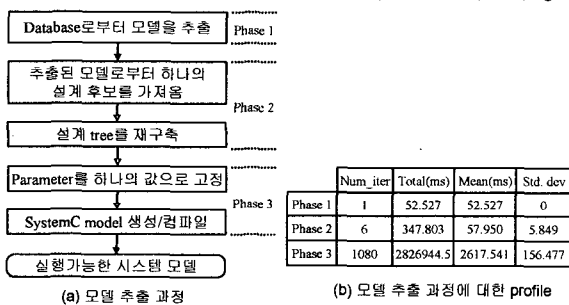


그림 5 모델의 추출 과정과 소요 시간

사용자의 개입 없이 자동적으로 수행된다. 그림 5(b)는 각 과정이 cache operation model에 대해서 몇 번이 수행되고 각 과정에서 얼마만큼의 시간이 소요되었는지를 보이고 있다. 실험은 Pentium IV 3GHz, 1GB main memory, Windows NT 환경에서 수행되었다.

5. 결론

본 연구에서는 모델의 재사용성을 높이기 위해 상위 레벨에서 모델을 설계하고 관리할 수 있는 기법을 제안하였다. 이를 위하여 attributed AND-OR graph라는 새로운 정형론을 제시하였다. 또한 모델의 확장과 추출이 자동적으로 이루어지도록 일련의 전체 과정을 실제로 구현

하여 실험하였다.

현재 SoC(System-on-a-Chip) 설계에서는 모델 재사용 방법론조차도 time-to-market을 만족시키기에 부족한 것으로 여겨지고 있다. 그 결과로 새로이 대두한 방법론은 platform-based design[5]으로 이는 모델 하나씩 재사용하는 것이 아니라 예전에 설계했던 시스템 전체를 초기 설계로 하여 조금씩 바뀌나감으로써 설계 시간을 단축시키는 것을 목표로 한다. 본 방법론을 확장하여 platform-based design 방법론까지 적용하는 것을 추후 과제로 한다.

참고문헌

- [1] Moore, G. "Progress in Digital Integrated Electronics", *IEEE International Electronic Devices Meeting*, 1975
- [2] John Zhong, "Synthesizable IP: The Risk Pays Off", *EE Times*, Dec. 19, 2002
- [3] <http://www.xml.org>
- [4] Bernard P. Zeigler, Herbert Prachofer, Tag Gon Kim, *Theory of Modeling and Simulation, 2nd edition*, Academic Press, 2000
- [5] Grant Martin, Henry Chang, *Winning the SoC Revolution: Experiences in Real Design*, Kluwer Academic Publishers, 2003