

논문 2013-50-1-20

Mixed Integer Linear Programming을 이용한 온칩 크로스바 네트워크 토폴로지 합성

(On-Chip Crossbar Network Topology Synthesis using
Mixed Integer Linear Programming)

전민제*, 정의영*

(Minje Jun and Eui-Young Chung)

요약

SoC내의 IP 개수 및 데이터 통신량이 증가함에 따라 온칩 크로스바 네트워크가 SoC의 중추 연결망으로서 널리 사용되어지고 있다. 온칩 크로스바 네트워크는 여러 개의 크로스바 스위치들과 이들간의 연결로 이루어지는데, 시스템의 복잡도가 증가함에 따라 IP들과 스위치들간의 연결 형태를 결정하는 것, 즉 토폴로지를 결정하는 것이 점점 복잡해지고 있다. 이 문제를 해결하기 위해 본 논문에서는 목표 시스템의 칩내 통신 특성을 고려하여 최적의 온칩 크로스바 네트워크의 토폴로지를 찾아주는 방법을 제안한다. 제안하는 토폴로지 합성 방법은 mixed integer linear programming(MILP)를 이용하여 다른 휴리스틱 합성 방법과 달리 전역 최적점(global optimum)을 찾을 수 있는 장점이 있다. 또한, 기존에 제안되었던 MILP를 이용한 토폴로지 합성 방법들이 토폴로지를 표현하는데 IP 노드들과 스위치들 간의 인접 행렬들을 이용했던 것과 달리, 본 논문에서는 IP들 간 통신을 표현하는 엣지들을 기본으로 하는 새로운 표현 방식을 제안한다. 실험 결과 본 논문에서 제안하는 새로운 MILP 표현 방식을 이용할 경우 기존 MILP 표현 방식을 이용했을 때보다 4개의 예제들에 대해 합성 속도가 평균 77.1 배 향상되었다.

Abstract

As the number of IPs and the communication volume among them have constantly increased, on-chip crossbar network is now the most widely-used on-chip communication backbone of contemporary SoCs. The on-chip crossbar network consists of multiple crossbars and the connections among the IPs and the crossbars. As the complexity of SoCs increases, it has also become more and more complex to determine the *topology* of the crossbar network. To tackle this problem, this paper proposes an on-chip crossbar network topology method for application-specific systems. The proposed method uses mixed integer linear programming to solve the topology synthesis problem, thus the global optimality is guaranteed. Unlike the previous MILP-based methods which represent the topology with adjacency matrixes of IPs and crossbar switches, the proposed method uses the communication edges among IPs as the basic element of the representation. The experimental results show that the proposed MILP formulation outperforms the previous one by improving the synthesis speed by 77.1 times on average, for 4 realistic benchmarks.

Keywords : 크로스바, 온칩 네트워크, 토폴로지, mixed integer linear programming

I. 서론

반도체 제조 기술의 발달과 다기능에 대한 시장의 요구로 인해 시스템 온 칩(SoC)에 내장되는 IP (intellectual property)의 개수가 끊임없이 증가해왔다. 이와 더불어 멀티미디어와 같은 데이터 집약적

* 정회원, 연세대학교 전기전자공학과
(Department of Electrical and Electronic Engineering,
Yonsei University)

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로
한국연구재단의 지원을 받아 수행된 연구임
(2012-0007181, 2011-0027625)

접수일자: 2012년10월22일, 수정완료일: 2013년1월3일

(data-intensive) 어플리케이션들은 칩내 IP들간 데이터 통신량을 폭발적으로 증가시켰다. 이러한 이유로 여러 IP들을 연결하고 이들간의 데이터 통신을 원활하게 하기 위한 온칩 연결망에 대한 연구가 최근 매우 활발히 진행되어왔다.

칩내 데이터 통신에 높은 대역폭이 요구됨에 따라 크로스바 스위치(crossbar switch)를 이용한 온칩 연결망은 이러한 요구에 따라 기존의 공유 버스 구조를 빠르게 대체해왔다. 하지만 시스템의 크기가 더욱 커짐에 따라 긴 와이어 딜레이와 크로스바 자체의 동작 속도 감소로 인해 여러 개의 분산된 크로스바 스위치를 이용한 온칩 크로스바 네트워크가 새로운 온칩 연결망으로 사용되기 시작하였다^[1] (그림 1).

온칩 연결망을 구성하는 크로스바의 개수가 증가함에 따라 각 크로스바의 크기, IP 및 다른 크로스바들과의 연결 형태를 결정하는 것, 즉, 크로스바 네트워크의 토폴로지를 결정하는 것이 매우 복잡한 문제로 대두되었다. 시스템의 크기가 커짐에 따라 ad-hoc 방식의 토폴로지 설계로는 한계가 있기 때문에 이를 해결하기 위한 토폴로지 자동 합성 방법들이 활발히 연구되었다^[1~6]. 하지만 이중 많은 방식들은 결과물의 전역 최적성(global optimality)이 보장되지 않는 휴리스틱 방법들이 대부분이다^[1~2, 4, 6].

사실 이러한 휴리스틱 방법들이 주로 연구된 이유는 토폴로지 설계의 디자인 스페이스가 매우 크기 때문이다. [3]에 소개된 방법에서는 전역 최적 솔루션을 찾기 위해 mixed integer linear programming(MILP)를 이용한 토폴로지 합성 방법이 제안되었지만 지나치게 긴 합성 시간이 문제점으로 지적되었다. 이 문제를 해결하기 위해 [4]에서는 [3]에서 제안된 MILP 방법과 휴리스틱

방법을 조합한 알고리즘이 제안 되었으나 다른 휴리스틱 방법들과 마찬가지로 전역 최적성은 보장되지 않는다.

본 논문에서는 전역 최적성을 유지하면서도 긴 합성 시간의 문제를 해결하기 위해 새로운 MILP 표현식에 기반한 토폴로지 합성 방법을 제안한다. [3~5]에서 사용된 MILP 표현식들은 토폴로지를 IP들과 스위치 노드들 간의 인접 행렬(adjacency matrix)들로 표현하는 접근 방법을 사용한 것과 달리, 본 논문에서는 IP들간의 통신을 표현하는 엣지(edge)들이 어떤 스위치들을 통해 연결되는가를 나타내는 행렬을 기본으로 한다. 실험 결과 본 논문에서 제안하는 새로운 MILP 표현식은 [3]에서 제안된 표현식 대비 평균 77.1배의 속도 향상을 보였다.

II. 본 론

1. 가정 및 문제 정의

가. 가정

본 논문은 기본적으로 [3]과 동일한 문제를 목표로 하되 더 나은 MILP 수식을 제안하여 수행 시간의 문제를 개선하는 것을 목적으로 한다. 따라서 [3]과 동일하게 다음과 같이 다음 사항들을 가정한다.

- 특정 마스터-슬레이브 간 통신은 단일 라우팅 경로를 통해 이루어진다 (single path routing).
- 모든 크로스바들은 동일한 클럭 주파수로 동작한다.
- 한 크로스바와 다른 크로스바는 하나의 링크로만 연결된다 (single inter-crossbar connection).
- 한 크로스바와 다른 크로스바의 연결에는 하나의 파이프라인 스테이지가 삽입된다.

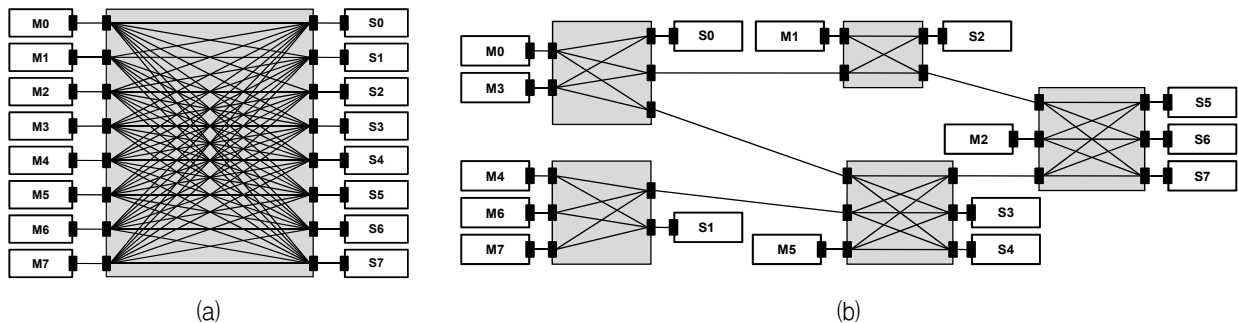


그림 1. 단일 크로스바 연결망(a)과 다중 크로스바들로 이루어진 크로스바 네트워크(b)의 예
Fig. 1. Examples of (a) a single crossbar network and (b) a cascaded crossbar network.

나. 문제 정의

본 논문에서는 IP들 간의 통신 특성이 주어진 시스템의 크로스바 네트워크 토폴로지 합성 문제를 다룬다. 제안하는 합성 방법은 다음 정보들을 입력으로 받는다:

- *Communication Requirement Graph (CRG)* $G(V_M, V_S, E)$ 는 무향 그래프(undirected graph)이다. $v_m \in V_M$, $v_s \in V_S$ 은 각각 마스터 인터페이스와 슬레이브 인터페이스를 갖는 포트를 의미한다. $e \in E$ 는 $src(e) \in V_M$ 와 $dst(e) \in V_S$ 사이의 데이터 통신을 의미한다. $BW_r(e)$, $BW_w(e)$, $LAT_r(e)$, $LAT_w(e)$ 는 그 통신에 요구되는 성능 특성들로서 각각 읽기 통신량, 쓰기 통신량, 읽기 지연시간, 쓰기 지연시간에 대한 요구치를 의미한다.
- 다양한 크기의 크로스바 스위치들에 대한 물리적 특성 정보를 담은 라이브러리. $A_{i,j}$ 와 $F_{i,j}$ 는 각각 i 개의 입력 포트(즉, 슬레이브 인터페이스 포트)와 j 개의 출력 포트(즉, 마스터 인터페이스 포트)를 갖는 크로스바 스위치의 면적과 최대 동작 주파수를 의미한다. A_{pipe} 는 파이프라인 스테이지(혹은 레지스터 슬라이스)의 면적을 의미한다.

제안하는 토폴로지 합성은 위 정보들을 입력으로 받아 CRG에 정의된 모든 통신 요구치들을 만족하면서 크로스바 네트워크의 면적이 최소화되는 토폴로지 $T(V_M, V_S, X, L)$ 를 찾는 것으로 정의될 수 있다. 여기서 $x \in X$ 는 네트워크 내 크로스바를, $l \in L$ 은 노드들 간의 물리적 링크를 의미한다.

한 편, [5]에서는 [3]에서 소개되는 MILP 방법을 확장하여 전력 소모 및 네트워크 컨텐션에 의한 지연 시간을 MILP로 표현하는 방법을 소개하였는데, 동일한 방식으로 본 논문에서 제안하는 MILP 표현식에 추가될 수 있다. 앞서 말했듯이 본 논문의 목적은 가장 빠대가는 부분, 즉 [3]에 소개되는 MILP 표현식을 개선하는 것이기 때문에 전력 소모와 네트워크 컨텐션에 의한 지연 시간은 고려하지 않는다. 마찬가지로 [4]에 소개된 다중 시나리오에 대한 고려 또한 동일하게 적용될 수 있으며 본 논문에서는 고려하지 않는다.

2. MILP Formulation

가. 기본 결정 변수(decision variable) 및 제약 조건 제안하는 방식은 $e \in E$ 가 어떤 크로스바 스위치 $x \in X$ 들에 할당되는가를 토폴로지 표현의 기본으로 삼는다. 이는 다음 행렬을 통해 표현된다.

- EX 는 boolean 변수를 갖는 2차원 행렬로 원소 $ex_{i,j}$ 는 엣지 $i \in E$ 가 크로스바 $j \in X$ 를 지난다면 1, 그렇지 않다면 0을 갖는다.

예를 들어 $X = \{x_1, x_2, x_3\}$ 이고 $ex_{i,x_1} = 1$, $ex_{i,x_2} = 1$, $ex_{i,x_3} = 0$ 이라면 엣지 i 는 크로스바 x_1 과 크로스바 x_2 를 지난다는 의미이다. 주의할 점은 크로스바를 나타내는 원소 $x_k \in X$ 에서 인덱스 k 가 데이터가 지나게 되는 순서를 의미한다는 것이다. 즉, 위의 예에서 엣지 i 는 $src(i) \rightarrow x_1 \rightarrow x_2 \rightarrow dst(i)$ 의 형태로 연결된다는 의미이다.

제약 조건 1: 하나의 엣지는 반드시 하나 이상의 크로스바에 연결되어야만 하기 때문에 다음 제약 조건이 성립되어야 한다.

$$\forall e \in E, \sum_{x \in X} ex_{e,x} \geq 1 \quad (1)$$

- RX 는 boolean 변수를 갖는 2차원 행렬로 원소 $rx_{i,j}$ 는 엣지 $i \in E$ 가 처음으로 연결되는 크로스바가 $j \in X$ 일 경우 1, 그렇지 않을 경우 0을 갖는다.

이러한 특성을 갖는 행렬 RX 는 다음 수식들을 통해 구해질 수 있다.

$$\forall e \in E, x_k \in X, \sum_{\substack{x_i \in X, \\ i < k}} ex_{e,x_i} + rx_{e,x_k} \leq 1 \quad (2)$$

$$ex_{e,x_k} - rx_{e,x_k} \geq 0 \quad (3)$$

수식 (2)은 임의의 엣지 e 가 하나 이상의 x_k 보다 작은 인덱스를 갖는 크로스바와 연결되었을 경우, 즉 x_k 에 앞서 다른 크로스바와 연결이 되었을 경우

rx_{e,x_k} 를 0으로 강제하는 역할을 한다. 수식 (3)는 엣지 e 가 다른 크로스바들과의 연결과 상관없이 x_k 에 연결되지 않았을 경우 (즉, $ex_{e,x_k} = 0$ 일 경우) rx_{e,x_k} 을 0으로 강제하는 역할을 한다. 수식 (2)과 (3)에 의해 RX 는 엣지 e 가 처음 지나게 되는 크로스바 x_k 를 제외한 다른 모든 원소들에 대해 0으로 강제된다. RX 가 원하는 포인트에서 1을 갖기 위해서는 다음 제약 조건이 필요하다.

제약 조건 2: 하나의 엣지가 최초로 연결되는 크로스바는 하나일 수밖에 없으므로 다음 제약 조건이 만족되어야 한다.

$$\forall e \in E, \sum_{x \in X} rx_{e,x} = 1 \quad (4)$$

수식 (4)에 의해 RX 는 (2)과 (3)에 의해 0으로 강제된 포인트를 제외한 지점, 즉 엣지 e 가 처음으로 지나가는 크로스바 x 에 대해 $rx_{e,x} = 1$ 을 갖게 된다.

제약 조건 3: 물리적으로 하나의 마스터 노드 v_m 은 하나의 크로스바와 연결될 수밖에 없기 때문에, 동일한 마스터 노드에 연결된 엣지들에 대해 이들이 처음 연결된 크로스바는 모두 같아야 한다.

$$\forall e_1, e_2 \text{ where } src(e_1) = src(e_2), x \in X, rx_{e_1,x} = rx_{e_2,x} \quad (5)$$

RX 행렬과 대칭되는 행렬로서 NX 행렬을 다음과 같이 정의할 수 있다.

- NX 는 boolean 변수를 갖는 2차원 행렬로 원소 $nx_{i,j}$ 는 엣지 $i \in E$ 가 마지막으로 지나게 되는 크로스바가 $j \in X$ 일 경우 1, 그렇지 않을 경우 0을 갖는다.

수식 (2), (3)와 대칭되는 역할로서 NX 에 대해서는 다음 수식들이 필요하다.

$$\forall e \in E, x_k \in X, \sum_{\substack{x_i \in X, \\ i > k}} ex_{e,x_i} + nx_{e,x_k} \leq 1 \quad (6)$$

$$ex_{e,x_k} - nx_{e,x_k} \geq 0 \quad (7)$$

또한 제약 조건 2과 3에 대칭되는 역할로서 다음 제약 조건들이 필요하다.

제약 조건 4: 하나의 엣지가 마지막으로 지나가는 크로스바는 하나일 수밖에 없으므로 다음 제약 조건이 만족되어야 한다.

$$\forall e \in E, \sum_{x \in X} nx_{e,x} = 1 \quad (8)$$

제약 조건 5: 물리적으로 하나의 슬레이브 노드 v_s 은 하나의 크로스바와 연결될 수밖에 없기 때문에, 동일한 슬레이브 노드에 연결된 엣지들에 대해 이들이 마지막으로 지나가는 크로스바는 모두 같아야 한다.

$$\forall e_1, e_2 \text{ where } dst(e_1) = dst(e_2), x \in X, nx_{e_1,x} = nx_{e_2,x} \quad (9)$$

나. 토폴로지 타당성 평가를 위한 변수 및 제약 조건 앞에서 설명된 변수들 및 제약 조건들은 EX 행렬이 물리적으로 네트워크 토폴로지를 표현하도록 만들어주는 역할을 한다. 여기서는 그렇게 표현된 네트워크의 타당성(feasibility)을 판단하기 위한 수식들을 소개한다.

- EL 은 boolean 변수를 갖는 3차원 행렬로 원소 $el_{i,j,k}$ 는 $i \in E$ 가 $j \in X$ 의 마스터 포트와 $k \in X$ 의 슬레이브 포트를 잇는 링크를 필요로 할 경우 1, 그렇지 않을 경우 0을 갖는다.

EL 행렬은 다음 수식들을 통해 구해질 수 있다.

$$\forall e \in E, x_i, x_j \in X (i < j), 0.5 \times (ex_{e,x_i} + ex_{e,x_j}) \geq el_{e,x_i,x_j} \quad (10)$$

1) $j = i + 1$ 인 경우.

$$ex_{e,x_i} + ex_{e,x_j} \leq 1 + el_{e,x_i,x_j} \quad (11-1)$$

2) $j > i + 1$ 인 경우,

$$el_{e,x_i,x_j} + \alpha \times \sum_{k=i+1}^{j-1} ex_{e,x_k} \leq 1 \quad (11-2)$$

$$ex_{e,x_i} + ex_{e,x_j} + (1 - \alpha \times \sum_{k=i+1}^{j-1} ex_{e,x_k}) \leq 2 + \beta + el_{e,x_i,x_j} \quad (11-3)$$

수식 (10)은 엣지 e 가 x_i 와 x_j 중 하나라도 지나지 않을 경우 el_{e,x_i,x_j} 를 0으로 강제한다. $j=i+1$ 일 경우 x_i 와 x_j 사이에 연결된 다른 크로스바가 있을 수 없기 때문에 $ex_{e,x_i} = ex_{e,x_j} = 1$ 이면 $el_{e,x_i,x_j} = 1$ 이 되어야 하고 이는 수식 (11-1)에 의해 강제된다. $j > i+1$ 인 경우 x_i 와 x_j 사이에 연결된 다른 크로스바들이 존재할 수 있기 때문에 수식 (11-1) 대신 수식 (11-2)와 (11-3)이 필요하다. 수식 (11-3)은 e 가 x_i 와 x_j 사이의 하나 이상의 다른 크로스바와 연결되었을 경우 el_{e,x_i,x_j} 를 0으로 강제한다. (11-2) 좌변 둘째 항의 α 는 $1/\max(1, j-i-2)$ 의 값을 갖는 상수로서 둘째 항의 절대값을 1이하로 만들어주는 역할을 한다. (11-3)은 e 가 x_i 와 x_j 를 모두 지나고 그 사이에 연결된 다른 크로스바를 지나지 않을 경우 el_{e,x_i,x_j} 을 1로 강제한다. (11-3) 우변의 β 는 $\alpha \times \max(0, j-i-3)$ 의 값을 갖는 상수로서 좌변의 $\sum_{k=i+1}^{j-1} ex_{e,x_k} = 1$ 일 경우에도 el_{e,x_i,x_j} 가 0이 될 수 있게 만들어준다.

EL 행렬을 이용하여 네트워크의 각 링크에 부과되는 통신량을 다음과 같이 구할 수 있다.

$$\forall e \in E, \quad rbw_{i,j} = \sum_{i \in X} \sum_{j \in X} (el_{e,i,j} \times BW_r(e)) \quad (12-1)$$

$$wbw_{i,j} = \sum_{i \in X} \sum_{j \in X} (el_{e,i,j} \times BW_w(e)) \quad (12-2)$$

$rbw_{i,j}$ 와 $wbw_{i,j}$ 는 각각 크로스바 i 의 마스터 포트와 j 의 슬레이브 포트를 잇는 링크에 부과되는 읽기 통신량과 쓰기 통신량을 나타내는 편의 변수(convenience variable)들이다.

제약 조건 6 (대역폭 제약 조건): 모든 링크들에 부과되는 통신량은 링크의 최대 이론적 최대 대역폭을 초과할 수 없다.

$$\forall i, j \in X, \quad \begin{aligned} rbw_{i,j} &\leq freq \times datawidth \\ wbw_{i,j} &\leq freq \times datawidth \end{aligned} \quad (13)$$

$datawidth$ 는 링크의 데이터폭을 의미하는 상수이다. $freq$ 는 네트워크의 클럭 주파수를 의미하는 결정 변수로서 이를 구하는 방법은 [3]에서 소개된 방법과 동일한 방법을 사용하였다.

제약 조건 7 (지연시간 제약 조건): 모든 엣지들에 대해 엣지가 지나는 경로의 지연 시간 총합은 CRG 에 주어진 지연 시간 제약을 초과할 수 없다.

$$\forall e \in E, \quad \begin{aligned} \sum_{x \in X} ex_{e,x} &\leq LAT_r(e) \times freq \\ \sum_{x \in X} ex_{e,x} &\leq LAT_w(e) \times freq \end{aligned} \quad (14)$$

다. 토폴로지의 비용 계산을 위한 변수 및 제약 조건 토폴로지의 비용(cost)를 계산하기 위해서는 사용된 각 크로스바의 슬레이브 포트 개수와 마스터 포트 개수를 계산해야한다. 이를 계산하기 위해 다음 변수를 추가로 정의해준다.

- XX 는 boolean 변수를 갖는 2차원 행렬로 원소 $xx_{i,j}$ 는 크로스바 i 의 마스터 포트가 크로스바 j 의 슬레이브 포트와 연결되었을 경우 1, 그렇지 않으면 0을 갖는다.*

x_i 와 x_j 사이에 링크를 필요로 하는 엣지가 하나라도 있을 경우 이 둘 사이에는 링크가 필요하므로 다음 수식으로 XX 행렬을 구할 수 있다.

$$\forall i, j \in X, \quad xx_{i,j} \geq \frac{1}{|E|} \times \sum_{e \in E} el_{e,i,j} \quad (15)$$

$$xx_{i,j} \leq \sum_{e \in E} el_{e,i,j} \quad (16)$$

* $j < i$ 인 경우 $xx_{x_i,x_j} = 0$ 으로 고정된다. [3]에서 증명되었듯이 이는 문제의 일반성(generality)를 해치지 않는다.

수식 (15)에서 $|E|$ 는 엣지의 총 개수이다.

크로스바의 입력 포트(즉, 슬레이브 포트)와 출력 포트(즉, 마스터 포트)는 IP의 마스터 포트 혹은 슬레이브 포트와 연결을 위한 것과 다른 크로스바와의 연결을 위한 것으로 나눌 수 있다. 크로스바 $i \in X$ 의 IP와의 연결을 위해 필요한 입력 포트 개수(pi_i^{IP})와 출력 포트 개수(po_i^{IP})는 다음 수식을 통해 구할 수 있다.

$$pi_i^{IP} = \sum_{e \in E} (rx_{e,i} \times edge_frac_master(e)) \quad (17)$$

$$po_i^{IP} = \sum_{e \in E} (nx_{e,i} \times edge_frac_slave(e)) \quad (18)$$

$edge_frac_master(e)$ 는 $src(e)$, 즉 엣지 e 가 연결된 마스터 IP에 연결된 엣지의 총 개수의 역수이다. 예를 들어 마스터 v_m 에 연결된 엣지가 e_1 과 e_2 일 경우, $edge_frac(e_1) = edge_frac(e_2) = 1/2$ 이다. e_1 과 e_2 가 모두 크로스바 i 를 처음 지날 경우 $rx_{e_1,i} \times 1/2 + rx_{e_2,i} \times 1/2 = 1$ 이 되어 엣지 e_1 과 e_2 가 크로스바 i 에 하나의 입력 포트를 추가하게 된다. 이와 유사하게 $edge_frac_slave(e)$ 는 $dst(e)$ 에 연결된 엣지의 총 개수의 역수이다.

다음으로 크로스바 x_i 가 다른 크로스바와의 연결을 위해 필요한 입력 포트의 개수($pi_{x_i}^{switch}$)와 출력 포트 개수($po_{x_i}^{switch}$)를 다음과 같이 구할 수 있다.

$$pi_{x_i}^{switch} = \sum_{j=1}^{i-1} xx_{x_j, x_i} \quad (19)$$

$$po_{x_i}^{switch} = \sum_{j=i+1}^{|X|} xx_{x_i, x_j} \quad (20)$$

최종적으로 크로스바 i 의 총 입력 포트의 개수(pi_i)와 총 출력 포트의 개수(po_i)는 다음과 같이 구해진다.

$$pi_i = pi_i^{IP} + pi_i^{switch} \quad (21)$$

$$po_i = po_i^{IP} + po_i^{switch} \quad (22)$$

이렇게 구해진 pi_i 와 po_i 를 이용해 크로스바 i 의 면적($area_i$)와 최대 동작 주파수($freq_i$)를 구하는 방법은

[3]에 소개된 방법과 동일한 방법을 사용 하였다. 또한 [3]과 마찬가지로 모든 크로스바와 파이프라인 스테이지 면적의 총합을 목적 함수(objective function)로 정의 하고 이를 최소화하는 것을 목적으로 하였다.

III. 실험

1. 실험 환경

평가를 위해 제안된 방법(e -MILP)과 [3]에 소개된 방법(n -MILP)을 표-1에 나열된 4개의 예제들에 대해 적용해보았다.* 제안하는 방법과 [3]에 소개된 방법 모두 동일한 문제를 다루고 전역 최적성이 보장되는 방법 들이기 때문에 얻어진 토폴로지의 결과물은 동일하다. 따라서 본 논문에서는 이 두 방법의 합성 시간만을 비교하도록 한다. 두 방법 모두 최대 크로스바의 개수는 5개로 고정하였다. 또한 n -MILP 방식은 최대 연결 깊이(cascading depth) N 을 입력으로 요구하는데 이는 최소값인 2로 설정하였다.**

크로스바 라이브러리는 AXI 프로토콜의 크로스바 스위치를 크기별로 합성하여 면적과 최대 동작 주파수를 얻었으며, 데이터폭은 64bit를 사용하였다.

e -MILP와 n -MILP 방법 모두 C++로 구현되었으며 Xpress-MP MILP solver^[8]를 이용하였다. 실험은 듀얼 코어 CPU와 1GB RAM을 갖는 가상 머신 상에서 진행 되었으며 호스트 머신은 쿼드 코어 2.8Ghz Core i5 CPU와 8GB RAM을 가진 x64 PC이다.

표 1. 실험에 사용된 벤치마크 명세
Table 1. Benchmark descriptions.

	$ V_M $	$ V_S $	$ E $	설명
mpeg4decode	9	3	12	MPEG4 디코더 ^[7]
multimedia_soc	12	4	21	멀티미디어 SoC ^[3]
mobile_mmp	12	5	20	모바일 멀티미디어 플레이어 ^[2]
mobile_ap	14	5	21	모바일 어플리케이션 프로세서 ^[2]

* e -MILP와 n -MILP의 e 와 n 은 각각 본 논문에서 제안하는 엣지 ('edge) 기반의 MILP 표현식을 사용한 것과 [3]의 방법이 노드 ('node) 기반의 표현식을 사용한 것을 상징한다.

** n -MILP 방식은 N 이 증가할수록 수행시간이 매우 급격히 증가한다. N 을 3으로 하였을 때 n -MILP는 mpeg4decode를 제외한 다른 벤치마크들에 대해 timeout deadline 안에 답을 찾지 못했다.

2. 실험 결과

표-2는 표-1에 기술된 벤치마크들에 대한 합성 결과를 보여준다. 먼저 표의 마지막 섹션인 ‘합성 시간 비교’ 부분을 살펴보면, 제안된 *e*-MILP가 *n*-MILP 보다 획기적으로 빠른 합성 속도를 갖는 것을 확인할 수 있다. mpeg4decode, multimedia_soc, mobile_mmp 벤치마크들에 대해 *e*-MILP는 최대 208.0배, 평균 77.1배의 속도 향상을 보였다. 또한, *n*-MILP의 경우 mobile_ap 벤치마크에 대해 24시간 안에 타당한(feasible) 토폴로지를 찾지 못한 반면, *e*-MILP는 약 6분(368.3초) 만에 합성을 마쳤다. 이러한 획기적인 속도 향상은 *e*-MILP의 효율적인 토폴로지 표현 방식(representation)에 기인한다. *n*-MILP의 경우 여러 단계 걸쳐 연결되는 네트워크를 표현하기 위해 4차원 이상의 다차원 변수 행렬들을 다수 도입하고 있다. 이로 인해 마스터 개수와 슬레이브 개수가 증가할수록 MILP 결정 변수와 개수를 급격히 증가하게 되고 결과적으로 문제 크기가 커질수록 합성 시간이 급격히 증가하게 된다. 표-2에서 문제 크기가 커질수록 *e*-MILP와 *n*-MILP의 시간 격차가 급격히 커지는 것이 이러한 사실을 반증한다.

한편 표-2의 ‘면적 비교’ 섹션에서는 제안하는 방법으로 합성된 크로스바 네트워크와 단일 크로스바의 면적 차이를 보여준다. 문제의 크기가 커질수록 합성된

네트워크의 면적 측면에서의 이득이 커지는 것을 확인할 수 있으며, 4개의 벤치마크들에 대해 최대 32.1%, 평균 19.8%의 면적 감소를 보였다.

IV. 결론

본 논문에서는 새로운 MILP 표현식을 이용한 크로스바 네트워크의 토폴로지 합성 방법을 제안하였다. 기존의 MILP 표현식이 마스터 노드, 슬레이브 노드, 크로스바 노드들의 연결을 나타내는 표현 방법을 사용한 것과 달리, 제안하는 방법에서는 옛지와 크로스바의 관계를 기반으로 한 새로운 접근 방법을 제시하였다. 실험 결과 제안된 방법은 기존 방법과 동일한 전역 최적 토폴로지를 찾아내면서 평균 77배 이상의 합성 속도 향상을 보였다. 본 논문에서는 비록 네트워크의 면적만을 다루었고 매우 단순한 지연 시간 모델을 사용하였지만, [3]에 기반하여 전력 소모와 보다 정확한 지연 시간 모델을 적용한 [5]의 방법에 쉽게 적용되어 그 성능을 크게 향상시킬 수 있을 것으로 예상된다.

표 2. 벤치마크들에 대한 합성 결과, 면적 감소 및 수행 시간 비교

Table 2. Synthesis results for the network area and the runtime.

mpeg4decode	multimedia_soc	mobile_mmp	mobile_ap
합성된 토폴로지			
면적 비교			
합성된 네트워크 면적 (mm ²) / 단일 크로스바 대비 면적 감소 비율 (%)			
0.475 / 0	0.630 / 19.3	0.672 / 27.9	0.739 / 32.1
합성 시간 비교			
<i>e</i> -MILP (A, 단위: 초) / <i>n</i> -MILP [3] (B, 단위: 초) / speedup (B/A)			
14.3 / 50.0 / 3.5	104.2 / 2046.7 / 19.7	267.5 / 55642.7 / 208.0	368.3 / timeout / n/a

* timeout deadline은 24시간(=86400초)

참 고 문 헌

- [1] J. Yoo, S. Yoo, and K. Choi, "Communication Architecture Synthesis of Cascaded Bus Matrix," *Proc. ASPDAC 2007*, pp. 171-177, Jan. 2007.
- [2] J. Yoo, S. Yoo, and K. Choi, "Topology/Floorplan/Pipeline Co-design of Cascaded Crossbar Bus," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1034 - 1047, Aug. 2009.
- [3] M. Jun, S. Yoo, and E. Y. Chung, "Mixed Integer Linear Programming-based Optimal Topology Synthesis of Cascaded Crossbar Switches," *Proc. ASPDAC 2008*, pp. 583-588, Jan. 2008.
- [4] M. Jun, S. Yoo, and E. Y. Chung, "Topology Synthesis of Cascaded Crossbar Switches," *IEEE Trans. on Computers-Aided Design of Integrated Circuits and Systems*, vol. 28, pp. 926-930, Jun. 2009.
- [5] Y. Jang, J. Kim, and C.M. Kyung. "Topology Synthesis for Low Power Cascaded Crossbar Switches." *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, issue 12, pp. 2041 - 2045, Dec. 2010.
- [6] M. Jun and E.Y. Chung. "Design of On-Chip Crossbar Network Topology using Chained Edge Partitioning." *The Computer Journal*, vol. 53, issue 7, pp. 904-917, Feb. 2010.
- [7] E.B.Van der Tol, E.G.T.Jaspers,"Mapping of MPEG-4 Decoding on a Flexible Architecture Platform", *SPIE 2002*, pp. 1-13, Jan, 2002.
- [8] FICO, [online] <http://www.fico.com>.

저 자 소 개



전 민 제 (정회원)
2006년 연세대학교 학사 졸업
2008년 연세대학교 석사 졸업
2013년 연세대학교 박사 졸업
<주관심분야 : SoC 구조 설계, 반도체 설계 자동화>



정 의 영 (정회원)
1988년 고려대학교 학사 졸업
1990년 고려대학교 석사 졸업
2002년 Stanford University
박사 졸업
<주관심 분야 : 시스템 구조, VLSI 설계, 저전력 설계>