

# 다중 버스 기반 SoC 구조의 정량적 통신 성능 분석

이재성 · 박재홍\*

한국교통대학교

## A Quantitative Communication Performance Analysis of Multi-Layered Bus-Based SoC Architectures

Jaesung Lee · Jae-Hong Park\*

Korea National University of Transportation

E-mail : jaesung.lee@ut.ac.kr, 1805pjh@naver.com (\*교신)

### 요 약

최근 SoC 업계에서는 다양한 다중 버스 구조가 사용되고 있다 그러나, 무분별한 버스 층의 남용은 통신 자원과 실리콘 면적의 낭비를 초래한다 본 논문은 이러한 낭비를 막기 위한 최적의 다중 버스 구조를 탐색하는 정량적 분석법을 소개한다 본 방법은 다양한 온칩 버스 프로토콜의 특성을 수학적 모델 형태로 반영하여 서로 다른 프로토콜을 기반으로 합성된 버스 구조간 비교가 가능하다 예제를 대상으로 실험한 결과 AHB, AXI, SNP 프로토콜 기반으로 합성된 다중 버스 구조 중 SNP 기반으로 합성된 버스 구조가 AXI 기반의 다중 버스 구조 대비 20% 더 성능이 좋으며 제안된 방법들을 통한 시간 복잡도도 상당히 저감된 것으로 확인되었다

### ABSTRACT

Recently, the SoC industry mainly uses various multi-layered bus architectures. However, reckless use of bus layers may results in on-chip communication resources and waste of silicon area. This paper performs a quantitative analysis to compare the two de-facto on-chip buses and SNP. Through the performance estimation, the performance of SNP turns out to be significantly enhanced for asymmetric write and read traffic (non-central F distribution) while symmetric traffic is similar to that of AXI. More specifically, SNP properly places IP cores on the top or bottom, induces the write and read channels to be balanced, and achieves about twenty percent improved performance compared to AXI.

### 키워드

SoC, 온칩 버스, SNP, 다중 버스 구조

### 1. 서 론

전통적으로 AMBA AHB [1], AXI [2] CORE CONNECT 그리고 STBUS와 같이 버스기반 INTERCONNECTS는 많은 IP코어 사이의 저렴한 비용으로 상호연결이 가능하기 때문에 널리 사용되어왔다. 그러나 버스에 다수의 IP 연결은 각 IP에 할당된 대역폭을 심각하게 감소시켜 성능 저하를 유발하고 버스 신호선의 수를 급격히 증가

시킬 수 있다. 그에 따라 mezzanine 또는 piggy-back 구조를 사용한 다중 버스 구조가 대 중화 되었다. 현재 다중 버스 구조는 긴 버스를 단순히 여러 마디로 나누어 연결한 segmented 구조로부터 각 IP 코어당 하나의 레이어를 생성해 각 IP 간 모든 연결을 지원하는 버스 매트릭스까지 다양한 형태로 진화하고 있다 그러나, 이러한 버스 레이어의 무분별한 사용은 과도한 버스 신호선수 낭비 뿐만 아니라 브릿지, 아비터 등

의 통신 자원의 낭비를 초래할 수 있다.

신호선 수의 과도한 사용을 피하기 위한 노력으로 SNP 가 개발되었다[4]. 일반적인 표준 버스는 그림 1에서 보는 바와 같이 주소, 컨트롤, 데이터쓰기, 데이터읽기용 버스들이 각각 분리된 신호의 형태로 발표되었다. 버스 아비터는 IP 전송간 상호배제를 관리하고자 중앙 멀티플렉서를 제어하여 전송 충돌이 없도록 한다. 그림의 버스 상단부에는 임의 숫자의 마스터 IP 들이 연결될 수 있고, 그림의 버스 하단부에는 임의 숫자의 슬레이브 코어들이 연결될 수 있다. 이 예제에서, 3개의 마스터와 3개의 슬레이브는 각각 'M'과 'S'으로 표시되었다. 성능적 측면에서 AHB는 상당히 제약 사항이 많았다. 왜냐하면 이전 전송이 완전히 완료되지 않으면 다음 전송을 시작하지 못하도록 프로토콜이 규정되어 있기 때문이다. 따라서, 쓰기 데이터 전송과 읽기 데이터 전송 중 오직 한번에 한 가지 타입의 전송만 가능하다 반면 AXI의 경우 전송이 진행중에 다른 전송을 시작할 수 있도록 허용하기 때문에 쓰기 데이터 버스와 읽기 데이터 버스를 각각 독립적으로 운영할 수 있다. 즉, 읽기와 쓰기를 동시에 전송하는 것이 가능하다.

그림의 좌측 화살표 기호는 그러한 AHB 특징을 상징적으로 나타내고 있으며(즉, 한 번에 한 방향 전송), 오른쪽에 있는 한 쌍의 화살표들은 이러한 AXI 특징을 상징적으로 나타낸 것이다(즉, 한 번에 두 방향 전송). 하지만, AXI는 여전히 주소, 컨트롤, 데이터쓰기, 데이터읽기 신호 버스를 따로 정의하여 사용하고 있어 과도한 신호선 문제는 그대로 남아 있다.

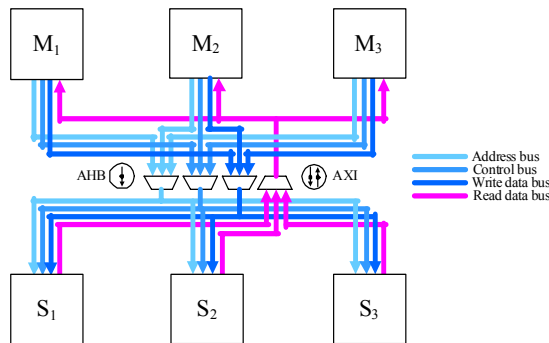


그림 1. AHB와 AXI와 같이 전형적인 온-칩 버스 프로토콜을 기반으로 한 버스 구조

과도한 와이어 사용을 피하기 위해 SNP [4]는 페이지 기반의 전송 방식을 취하고 주소, 컨트롤, 데이터 정보 등이 상호 공유하는 채널을 통해 전송을 하기 때문에 매우 적은 수의 버스 신호를 사용한다. 추가적으로 3 비트 페이지 신호를 사용하여 채널을 통해 전송되는 신호가 어떤 신호인지를 구별할 수 있도록 하였고 잘 정의된 페이지 전송 규칙은 페이지 생략 및 복원을 가능케하여 통신 최적화를 구현한다. 결과적으로, SNP의 성

능은 AXI와 AHB의 신호선수 대비 대략 3/5 수준의 신호선만으로 비슷한 전송 성능을 나타내었다. SNP가 그런 성능에 도달할 수 있는 주요 이유는 첫째, 기존 각각 신호 그룹을 별도로 정의는 하였지만 대부분 메모리 및 레지스터 파일은 한 사이클 전에 주소신호를 받은 후 그 다음 사이클에서 데이터를 읽어들이기 때문에 어차피 2사이클이 필요하였으며 둘째, 반복적으로 전송되는 주소 정보와 제어 정보는 종종 불필요하며 버스 대역폭만 잡아먹으므로 생략해도 무방하며 셋째 페이지 인터리빙을 지원하여 AXI처럼 동시 쓰기와 읽기 전송을 허용하기 때문이다.

특히, SNP는 동일한 모양을 채널을 Up channel과 Down channel에 사용하기 때문에 마스터/슬레이브 구분없이 IP 코어들을 버스의 상단 하단 어느 곳에나 붙일 수 있다.

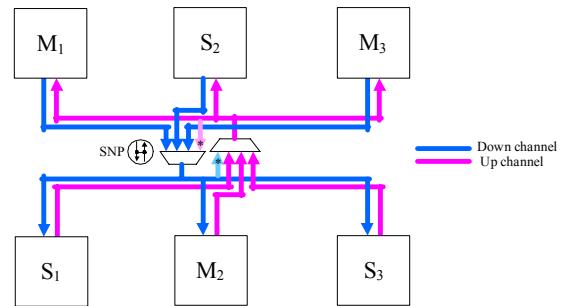


그림 2. SNP를 기반으로 한 싱글 버스구조

그림 2에서 보이는 것처럼 두 세트의 동일한 SNP 채널 신호는 각 IP 코어에 대해 대칭적인 통신 경로를 형성한다. 따라서, 기존의 버스와는 달리 각 코어는 마스터/슬레이브 구분없이 상단 또는 하단 어디에도 연결될 수 있다. 만약 쓰기 데이터량과 읽기 데이터량간 차이가 크면 적절한 위치에 각 IP코어를 배치해 대역폭을 조절하여 전송 성능을 향상시키는 게 가능하다.

논문의 나머지는 다음과 같이 구성된다. 제 2절에서는 세 버스 프로토콜별 다중 버스에 대한 정량적 분석법을 소개하고 제 3절에서는 본 분석법으로 버스 프로토콜들간 성능을 비교한 후 제 4절에서 결론을 내린다.

## II. 정량적 분석

본 절에서는 싱글 버스에 대한 통신 시간 비용 계산식을 세우고 이를 바탕으로 다중 버스 구조에 대한 통신 시간 비용 계산식을 완성한다.

### A. 싱글 버스 구조에서 통신 시간 비용식

표 1에서 주어진 IP 코어의 한 쌍 사이 데이터 트래픽을 가정하면 마스터와 슬레이브는 'M'과 'S'로 각각 나타내어지고 각 마스터 당 표에는 2

개의 세로 열이 존재한다. 왼쪽 열은 마스터에서 슬레이브로 쓰기 데이터 트래픽(즉, 그림1의 하향 트래픽)량을 나타내고 슬레이브로부터 마스터를 향하는 읽기 데이터 트래픽(즉, 그림1의 상향 트래픽)량은 두번째 열에 표시된다. 제 1 절에서 언급한 바와 같이 AHB는 오직 한번에 한방향 전송을 허용하므로 통신 시간 비용은 트래픽 타입에 관계없이 모든 셀들 값의 합계가 된다. 즉, 표1에서 AHB의 통신 시간 비용은 다음과 같이 계산된다.

$$CT_{AHB} = \sum W_{M_i, S_j} + \sum R_{M_i, S_j} = 30$$

표 1. 통신 표(Communication Table)

|                | M <sub>1</sub> |   | M <sub>2</sub> |   | M <sub>3</sub> |   |
|----------------|----------------|---|----------------|---|----------------|---|
| S <sub>1</sub> | 7              | 1 | 1              | 1 | 3              | 1 |
| S <sub>2</sub> | 0              | 0 | 10             | 1 | 0              | 0 |
| S <sub>3</sub> | 2              | 1 | 0              | 0 | 1              | 1 |

마스터 M<sub>i</sub>에서 슬레이브 S<sub>j</sub> 그리고 R<sub>M<sub>i</sub>,S<sub>j</sub></sub>로부터 전송된 쓰기 트래픽의 양을 나타내는 W<sub>M<sub>i</sub>,S<sub>j</sub></sub>는 슬레이브 S<sub>j</sub>에서 마스터 M<sub>i</sub>로부터의 읽기 트래픽을 나타낸다.

AXI에선, 쓰기 데이터 버스와 읽기 데이터 버스는 각각 관리되므로 시간 비용은 쓰기 트래픽 비용과 읽기 트래픽 비용 값 중 큰 값으로 결정된다. 따라서 다음과 같이 계산된다.

$$CT_{AXI} = \max(\sum W_{M_i, S_j}, \sum R_{M_i, S_j}) = 24$$

한편, SNP에서의 계산은 복잡하다. AHB와 AXI의 경우와 달리 IP 코어들을 버스 상단 하단 어느 곳에나 붙일 수 있기 때문에 각 IP코어의 부착 위치에 따라, 다양한 계산 결과가 나올 수 있다. 그림 1에서 M<sub>2</sub>와 S<sub>2</sub>를 그림2에서 보는 바와 같이 위 아래를 서로 바꾸어 연결하면 표 2와 같이 표를 변환해야 한다. 예를 들어 M<sub>1</sub> 또는 M<sub>3</sub>이 S<sub>1</sub> 또는 S<sub>3</sub>에 데이터를 쓸 때 DOWN 채널에서 회귀되는 전송로(U-turn Path)가 발생된다. 한편, M<sub>2</sub>가 S<sub>2</sub>와 같은 방식으로 통신을 할 때 반대 채널에서 그런 현상이 발생하게 된다. 반대 채널에서 M<sub>2</sub>가 S<sub>2</sub>와 같은 작업을 할 때 발생하는 반면, S<sub>1</sub> 또는 S<sub>3</sub>에서 데이터를 읽어들이는 M<sub>1</sub>, M<sub>3</sub>는 UP 채널에 그러한 현상이 발생된다. 이렇게 같은 쪽에 위치한 IP 간 통신에 의한 U-turn Path가 사용되는 경우는 AHB 형태의 통신이 이루어지고 서로 다른 위치의 IP 간 통신에서는 AXI 형태의 통신이 이루어지기 때문에 수정된 계산 방법이 필요한 것이다.

표 2의 경우, M<sub>2</sub>와 S<sub>2</sub>를 서로간 위치를 바꾸었기 때문에 W/R 통신표는 새로운 표(상단)로 수정이 되며 이를 기반으로 계산을 수행하면 하

단의 표가 완성되고 이를 이용해 SNP에 대한 시간 비용, 즉 CT<sub>SNP</sub>는 다음과 같이 계산이 된다.

$$CT_{SNP} = \max(\sum D_{M_i, S_j}, \sum U_{M_i, S_j}) = 16$$

표 2. D/U 통신 표

|                | M <sub>1</sub> |   | M <sub>2</sub> |    | M <sub>3</sub> |   |
|----------------|----------------|---|----------------|----|----------------|---|
| S <sub>1</sub> | 7              | 1 | 1              | 1  | 3              | 1 |
| S <sub>2</sub> | 0              | 0 | 1              | 10 | 0              | 0 |
| S <sub>3</sub> | 2              | 1 | 0              | 0  | 1              | 1 |

|                | M <sub>1</sub> |   | M <sub>2</sub> |       | M <sub>3</sub> |  |
|----------------|----------------|---|----------------|-------|----------------|--|
| S <sub>1</sub> | 7              | 1 | 1 + 1          | 3     | 1              |  |
| S <sub>2</sub> | 0 + 0          |   | 10 ↔ 1         | 0 + 0 |                |  |
| S <sub>3</sub> | 2              | 1 | 0 + 0          | 1     | 1              |  |

본 계산식의 결과로부터 쓰고 읽기 트래픽의 양에서 큰 차이가 있을 때 IP 코어를 재배열함으로써 인해 성능이 상당히 향상되었다는 것을 알 수 있다.

### B. 다중 버스 구조의 통신 시간 비용식

전체적인 통신 시간 비용을 최소화하기 위해 각각의 IP 코어를 적당한 레이어에 할당하는 방법에 대한 논의가 먼저 필요하다.

예를 들어, 그림 3과 같이 임의의 토폴로지를 갖는 다중 버스가 있을 때 두 개의 인접한 레이어간의 통신(inter)은 그들 사이의 브릿지를 통하여 수행된다. 반면 하나 이상의 레이어를 건너 뛰어 통신하는 두 레이어간 통신(over)은 그 경로에 위치해 있는 모든 브릿지를 통해 통신이 수행된다. 단일 레이어내의 통신(intra)은 브릿지를 통한 통신은 필요없지만 해당 레이어를 점유하기 때문에 그 동안 다른 통신에 사용할 수 없다.

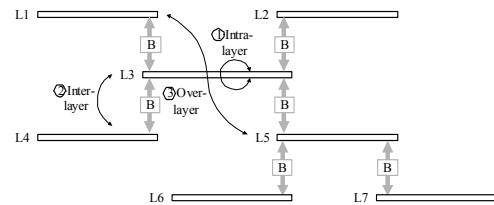


그림 3. 임의의 토폴로지를 갖는 multi-layer 버스의 시간 비용 고려

이러한 내용을 정리하여 다중 버스 구조에서 임의의 레이어에서의 통신 시간 비용은 다음3가지 요인에 의해 결정된다.

$$CT(l) = ct_{intra}(l) + ct_{inter}(l) + ct_{over}(l)$$

이 식은 임의의 한 레이어 L 에 대한 것이며 레이어 자체 내부 통신인 (1) intra-layer 통신 비용(즉, 두 IP코어 사이에 둘다 같은 레이어에 속함), 현재 레이어와 다른 레이어 간 통신인 (2) intra-layer 통신 비용, 본 레이어를 통해 이뤄지는 다른 두 레이어간 통신(즉, 상호 통신하는 두 개의 IP 코어 가 본 레이어에 속하진 않지만 통신 경로 상 반드시 본 레이어를 통과해야 하는 경우)인 (3)intra-layer 통신 시간 비용으로 구성된다. 이런 방식으로 다중 버스 구조의 모든 레이어  $L(1 < L < n, n$ 은 레이어의 총 수)에 대해 본 수식을 적용하여 계산하였을 때 나오는 최대값이 최소가 되도록 IP 들을 다중 버스 구조의 각 레이어에 배치한 구조가 최적의 구조가 되며 그 최소가 된 최대값이 본 다중 버스 구조의 성능이 된다

### III. 성능 평가

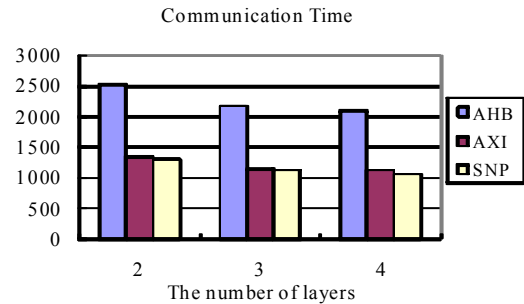
앞 절의 수식을 소프트웨어 프로그램으로 구현 후 몇 가지 예제 통신표들에 대해 실행하였다. 준비된 예제 통신표들은 IP코어의 수 12, 레이어수 2 ~ 3 개로 가정하여 합성 후 성능 값(통신 시간 비용)을 얻었다. 표의 값은 매트랩의 통계 도구를 사용하여 5개 표 한 세트에 균등분포를 따르도록 하여 무작위 발생하였다. 또한 다른 한 세트는 정규분포로, 또 다른 한 세트는 non-centric F 분포로 생성하였다. 이처럼 합성 데이터 값이 사용된 이유는 한 두가지 실 응용 사례만으로 평가할 경우 일반화의 오류에 빠질 가능성이 있기 때문이며 또한 여러 사례의 실제 응용을 저자가 구현해 보기에는 한계가 있기 때문이다 따라서, 현실적으로 이러한 방법이 가능한 많은 경우를 조사하는데 합리적인 방법이다

균등 분포의 경우 SNP는 쓰기 트래픽의 양과 읽기 트래픽 둘 다 비슷한 값을 가지게 되어 채널이 거의 대칭적으로 사용되기 때문에 AXI 와 거의 유사한 성능 값(통신 시간 비용 값)을 보였다. 반면, 정규분포를 따르는 경우 그림 4(a)와 같이 AHB 가 가장 저조한 성능이고 SNP 는 AXI 보다 다소 개선된 성능을 보였다. 한편 non-centric F 분포로 합성된 통신 표의 경우 쓰기와 읽기 데이터량간 큰 폭의 차이를 보임에 따라 그림 4 (b)에서 보이는 바와 같이 SNP의 성능이 AXI에 비해 좀 더 좋아짐을 알 수 있었다.

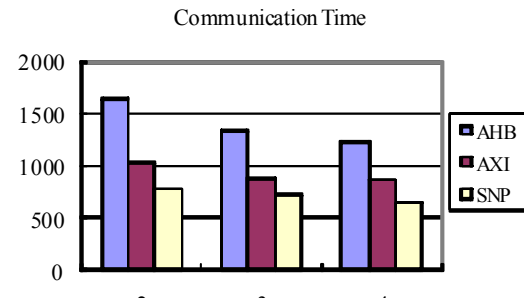
### IV. 결 론

이 논문은 기존의 2가지 업계 표준 온칩 버스와 SNP를 정량적으로 비교하는 방법을 제안하고 여러 다양한 통신 시나리오를 대상으로 시뮬레이션 분석을 수행하였다. 그 결과, 읽기 쓰기간 대칭 트래픽(균등분포 또는 정규분포)의 경우 SNP 는 AXI 에 비해 별다른 성능 개선을 보이지 않았지만 비대칭 쓰기와 읽기 트래픽(non-centric F

분포)의 경우 AXI 에 비해 SNP가 약 20% 의 성능 향상을 보였다. 좀 더 구체적으로 설명하자면, SNP 는 IP 코어를 마스터/슬레이브 구분없이 버스 상단 또는 하단에 적절히 배치하여 쓰기와 읽기 채널이 균형 잡히도록 함으로써 성능을 개선시킬 수 있는 것이다.



(a) 정규 분포



(b) non-centric F 분포

그림 4. 3가지 버스 프로토콜간 통신 시간 비용 비교

\* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2012R1A1A1038515)

### 참고문헌

[1] ARM, *AMBA specification*, Rev. 2.0, 1999.  
 [2] ARM, *AMBA AXI protocol specification*, 2003.  
 [3] S. Pasricha, N. Dutt, E. Bozorgzadeh, and M. Ben-Romdhane, "Floorplan-aware automated synthesis of bus-based communication architectures," in *Proc. DAC*, pp. 565-570, June 2005.  
 [4] J. Lee and H.-J. Lee, "Wire Optimization for Multimedia SoC and SiP Designs," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 8, pp. 2202-2215, Sept. 2008.  
 [5] T. Seceleanu, V. Leppanen, J. Suomi, and O. Nevalainen, "Resource Allocation Methodology for the Segmented Bus Platform," in *Proc. Int'l Workshop on IP-based SoC Design*, pp. 394-394, Oct. 2002.