

# 저 전력 시스템 온 칩 설계를 위한 버스 분할 기술

## (Bus Splitting Techniques for Low Power SoC Design)

임 호 영 \*    윤 미 선 \*    신 현 철 \*\*    박 성 주 \*\*  
 (Hoyeong Lim)    (Misun Yoon)    (Hyunchul Shin)    (Sungju Park)

**요 약** 일반적으로 버스에서 소모되는 에너지는 전체 시스템에서 큰 비중을 차지한다. 버스 분할 방법은 시스템의 소모 에너지를 줄이고 각 버스 세그먼트들의 기생 부하(parasitic load)를 감소시킴으로써 지연시간을 줄이는데 사용될 수 있다. 버스를 분할함에 있어서 버스에 의해 상호 연결된 처리소자들 사이의 데이터 교환 확률 분포에 따라 가장 적은 에너지를 소모하는 버스 분할 방법은 달라질 수 있다.

본 연구에서는 수평적인 버스 분할 구조의 확장된 개념으로 트리구조 기반의 버스 분할 방법에 대해 연구하고, 이를 바탕으로 여러 가지 버스 분할 방법 중에서 주어진 시스템의 처리소자 간 데이터 교환 확률의 분포에 따라 가장 적합한 구조를 선택하는 문제에 대해 논하였다. 실험 결과는 제안된 방법들이 버스에서 소모되는 에너지를 최대 83%까지 감소시킬 수 있음을 보여준다.

**키워드** : 버스, 분할, 저전력

**Abstract** In general, bus system consumes a very significant portion of power in a chip. Bus splitting can be used to reduce the energy dissipation and to reduce the propagation delay on the bus by lowering the parasitic load of each bus segment. Data exchange probability distribution between a set of interconnected processing elements affects the average energy dissipation of the splitted bus architectures.

In this research, we have developed tree-based bus splitting techniques and design methodologies, as an extension of horizontally aligned bus splitting. We have developed the methodology to select near-optimal bus architectures for low energy dissipation when data exchange probability distribution of a system is given. Experimental results show that the proposed techniques can reduce energy dissipation on the bus by up to 83%.

**Key words** : Bus splitting, Low power design

### 1. 서 론

최근의 많은 VLSI 설계에서는 고성능을 만족시키면서도 소모 전력을 줄일 수 있는 기술이 요구되고 있다. 이는 이동전화나 노트북 컴퓨터와 같은 휴대용 제품이 보편화되면서 한정된 배터리 성능을 고려한 저 소모 전력에 대한 요구가 시스템의 경량화, 고성능화에 못지않게 증가하고 있기 때문이다[1]. 근래에 한 칩(chip)에 시스템이 구현되는 시스템 온 칩(System on Chip, SoC)이 많이 사용되고 있다. 이러한 회로 설계에 있어서, 한 칩 내의 처리소자(Processing Element, PE) 간의 데이

타 통신을 위해 사용되는 버스에서의 소모 전력은 전체 소모 전력에서 큰 비중을 차지하고 있다.

버스에서 소모되는 전력을 줄이기 위해서 1차원적이고 수평적인 버스 분할하는 방법[2], 페스 트랜지스터를 이용하여 버스를 여러 부분의 버스 세그먼트로 나누는 방법[3] 등 버스 분할 방법들이 연구되었다.

본 연구에서는 일반적인 버스를 분할하는 문제를 정의하고, 저 에너지 소모를 위한 트리구조 기반의 버스 분할 기법을 소개하였다. 또한 소모 전력을 줄이기 위해서 시스템내의 PE간의 데이터 교환 확률 분포가 버스 분할에서 적합한 구조의 버스 선택에 미치는 영향을 설명하였다.

2절에서는 현재까지의 버스 분할 방법들에 대해 기술한다. 3절에서는 새로운 트리구조 기반의 버스 분할 방법을 소개하고 이를 바탕으로 데이터 교환 확률 분포에 따른 버스 분할 방법에 대해 설명한다. 4절에서는 실험 결과를 기술하고, 마지막으로 5절에서는 결론을 맺는다.

\* 비 회 원 : 한양대학교 전자전기 제어계측공학과  
 hylim@digital.hanyang.ac.kr  
 msyoon@digital.hanyang.ac.kr

\*\* 종 신 회 원 : 한양대학교 전자컴퓨터공학과 교수  
 shin@hanyang.ac.kr  
 parksj@mslab.hanyang.ac.kr

논문접수 : 2003년 6월 9일  
 심사완료 : 2005년 5월 31일

### 2. 기존의 버스 분할 방법

많은 시스템에서 시스템 내의 재사용이 가능한 제어 패스와 데이터패스를 제공하기 위해 버스를 사용하고 있다[4]. 일반적으로 버스를 통해 전달되는 데이터의 양은 많기 때문에 전체 에너지 소모를 감소시키기 위해서는 버스에서 소모되는 전력을 줄이는 것이 효과적이다.

단순한 공유(shared) 버스에서는 모듈이나 처리소자(PEs)들이 공유 버스 선에 연결되어 있다. 일반적으로 버스 선은 큰 기생(parasitic) 저항과 커패시턴스를 갖고 있기 때문에 버스 선에서의 충전과 방전 과정에 의한 에너지 소모가 상당하다. 예를 들어, Alpha 21064와 Intel 80386의 버스 선에서 소모되는 에너지는 각각 전체 소모 에너지의 15%-30%를 차지한다[3].

이에 따라 전체 소모 에너지를 최소화하기 위해 버스를 분할하는 방법이 연구되었고, 이러한 버스 분할 문제는 NP-complete 문제로 알려져 있다[2].

실제 모듈이나 처리소자들은 2차원적인 공간에 배치되므로 트리구조 기반의 버스가 수평적인 버스구조에 비해 적합하다. 그러나 트리구조 기반의 버스에 대해서는 충분한 연구가 이루어지지 않았다. 수평적인 버스구조는 트리구조 기반의 버스의 한 형태라고 할 수 있다. 본 논문에서는 이러한 새로운 트리구조 기반의 버스 분할 방법을 소개하고, 시스템의 저 전력 설계를 위하여 처리소자 사이의 데이터 교환 확률 분포에 따른 효과적인 버스 분할 방법에 대해 기술하였다.

### 3. 새로운 버스 분할 방법

버스 분할을 통해 나누어진 로컬(local) 버스들은 단순한 공유 버스에 비해 버스 선의 길이가 줄어들기 때문에 각 로컬 버스의 기생 부하(parasitic load)가 감소하게 된다. 이에 따라 로컬 버스 당 물리적인 커패시턴스가 줄어든다. 같은 로컬 버스로 묶인 처리소자 사이에 데이터 교환이 이루어질 때, 다른 로컬 버스에는 신호 전이가 발생하지 않으므로, 단순한 공유 버스에 비하여 에너지 소모는 크게 줄어든다. 그림 1에서는 이러한 수평적인 버스구조와 트리구조 기반의 버스의 한 예를 나타낸다.

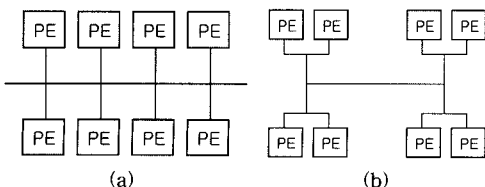


그림 1 (a) 수평적인 공유 버스구조 (b) 트리구조 기반의 버스구조

이 절에서는 단순한 공유 버스 구조와 수평적인 버스 분할 구조, 새로운 트리구조 기반의 버스 분할 방법과 실험에 사용한 PE간의 데이터 확률분포에 대해 기술하고, 버스 분할에 사용되는 버퍼를 모델링하였다.

PE간의 데이터 교환 확률 분포 매우 다양할 수 있다. 본 논문에서는 데이터 교환확률이 버스 분할에 미치는 영향을 실험적으로 분석하기 위하여, 표 1, 표 2, 표 3, 표 4, 표 5, 표 6과 같은 확률 분포의 표준편차(이하  $\sigma$ )가 다른 6가지의 실험 데이터를 사용하였다. 8개의 PE를 사용하였고, 표 1은  $\sigma=0.00$ , 표 2는  $\sigma=0.02$ , 표 3은  $\sigma=0.04$ , 표 4는  $\sigma=0.06$ , 표 5는  $\sigma=0.08$ , 표 6은  $\sigma=0.12$ 인 경우를 제시하였다.

끝으로 데이터 교환 확률 분포가 여러 가지 버스 분할 구조에 어떠한 영향을 미치는지 연구하였다.

표 1  $\sigma = 0.00$  인 경우

	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.0	0.035	0.035	0.036	0.036	0.036	0.036	0.036
P2	0.035	0.0	0.035	0.036	0.036	0.036	0.036	0.036
P3	0.035	0.035	0.0	0.035	0.036	0.036	0.036	0.036
P4	0.036	0.036	0.035	0.0	0.036	0.036	0.035	0.036
P5	0.036	0.036	0.036	0.036	0.0	0.035	0.036	0.036
P6	0.036	0.036	0.036	0.036	0.035	0.0	0.036	0.035
P7	0.036	0.036	0.036	0.035	0.036	0.036	0.0	0.035
P8	0.036	0.036	0.036	0.036	0.036	0.035	0.035	0.0

( $\sigma$ : 표준편차)

표 2  $\sigma = 0.02$  인 경우

	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.0	0.025	0.022	0.013	0.023	0.066	0.024	0.049
P2	0.025	0.0	0.022	0.013	0.006	0.024	0.029	0.015
P3	0.022	0.022	0.0	0.043	0.031	0.050	0.028	0.023
P4	0.013	0.013	0.043	0.0	0.049	0.070	0.027	0.029
P5	0.023	0.006	0.031	0.049	0.0	0.031	0.032	0.052
P6	0.066	0.024	0.050	0.070	0.031	0.0	0.035	0.084
P7	0.024	0.029	0.028	0.027	0.032	0.035	0.0	0.085
P8	0.049	0.015	0.023	0.029	0.052	0.084	0.085	0.0

( $\sigma$ : 표준편차)

표 3  $\sigma = 0.04$  인 경우

	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.0	0.148	0.001	0.001	0.012	0.002	0.006	0.092
P2	0.148	0.0	0.003	0.005	0.009	0.024	0.029	0.002
P3	0.001	0.003	0.0	0.043	0.031	0.097	0.048	0.006
P4	0.001	0.005	0.043	0.0	0.002	0.085	0.027	0.009
P5	0.012	0.009	0.031	0.002	0.0	0.009	0.009	0.052
P6	0.002	0.024	0.097	0.085	0.009	0.0	0.043	0.111
P7	0.006	0.029	0.048	0.027	0.009	0.043	0.0	0.094
P8	0.092	0.002	0.006	0.009	0.052	0.111	0.094	0.0

( $\sigma$ : 표준편차)

표 4  $\sigma = 0.06$  인 경우

	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.0	0.100	0.002	0.007	0.008	0.005	0.002	0.001
P2	0.100	0.0	0.003	0.130	0.021	0.012	0.010	0.002
P3	0.002	0.003	0.0	0.140	0.019	0.014	0.002	0.014
P4	0.007	0.130	0.140	0.0	0.002	0.019	0.014	0.012
P5	0.008	0.021	0.019	0.002	0.0	0.002	0.015	0.208
P6	0.005	0.012	0.014	0.019	0.002	0.0	0.102	0.014
P7	0.002	0.010	0.002	0.014	0.015	0.102	0.0	0.200
P8	0.001	0.002	0.014	0.012	0.208	0.014	0.200	0.0

( $\sigma$ : 표준편차)

표 5  $\sigma = 0.08$  인 경우

	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.0	0.179	0.002	0.007	0.001	0.004	0.002	0.001
P2	0.179	0.0	0.003	0.005	0.021	0.002	0.009	0.002
P3	0.002	0.003	0.0	0.147	0.009	0.004	0.002	0.006
P4	0.007	0.005	0.147	0.0	0.002	0.007	0.004	0.002
P5	0.001	0.021	0.009	0.002	0.0	0.002	0.008	0.312
P6	0.004	0.002	0.004	0.007	0.002	0.0	0.002	0.004
P7	0.002	0.009	0.002	0.004	0.008	0.002	0.0	0.251
P8	0.001	0.002	0.006	0.002	0.312	0.004	0.251	0.0

( $\sigma$ : 표준편차)

표 6  $\sigma = 0.12$  인 경우

	P1	P2	P3	P4	P5	P6	P7	P8
P1	0.0	0.002	0.008	0.002	0.001	0.005	0.002	0.002
P2	0.002	0.0	0.001	0.002	0.001	0.489	0.009	0.001
P3	0.008	0.001	0.0	0.005	0.001	0.001	0.001	0.001
P4	0.002	0.002	0.005	0.0	0.447	0.002	0.001	0.006
P5	0.001	0.001	0.001	0.447	0.0	0.002	0.001	0.001
P6	0.005	0.489	0.001	0.002	0.002	0.0	0.003	0.002
P7	0.002	0.009	0.001	0.001	0.001	0.003	0.0	0.001
P8	0.002	0.001	0.001	0.006	0.001	0.002	0.001	0.0

( $\sigma$ : 표준편차)

3.1 트리구조 기반의 버스 분할 방법

전체  $n$  개의 처리소자들이 하나의 공유 버스에 연결되어 있다고 가정하자.  $exchange(P_i, P_j)$ 는 처리소자  $P_i$ 와  $P_j$ 가 데이터를 교환할 확률을 나타내고,  $C_{BUS}(P_i, P_j)$ 는 처리소자  $P_i$ 와  $P_j$ 를 연결하는 버스의 커패시턴스일 때, 소모전력 비용함수는 식 (1)의  $P_{cost}$ 에 비례한다.

$$P_{cost} = \sum_{i \neq j} C_{BUS}(P_i, P_j) \cdot exchange(P_i, P_j) \quad (1)$$

저 전력에 대한 요구를 만족시키기 위해서 버스 선에서의 에너지 소모를 고려해야 한다.  $C_{BUS}$ 는 버스에서의 부하 커패시턴스이고,  $V_D$ 는 전압의 변화량이라고 할 때, 커패시티브 노드에서의 충전과 방전에 의한 평균 소모 에너지는 식(2)로 표현 할 수 있다.

$$E = 0.5 \times sw \times C_{BUS} \times V_D^2 \quad (2)$$

여기서  $sw$ 는 평균적인 switching activity,  $C_{BUS}$ 는 버스 선상의 커패시턴스를 의미한다. 평균 소모 에너지는 일정한  $V_D$  조건 하에서  $C_{BUS}$ 에 비례한다.

커패시턴스  $C_{BUS}(P_i, P_j)$ 는  $P_i$ 와  $P_j$ 를 연결하는 버스 구조에 따라 변할 수 있다. 각 처리소자간의 데이터 통신 확률이 주어졌을 때, 에너지 소모를 최소화하기 위해서는  $P_{cost}$ 값을 최소화시킬 수 있는 버스구조를 찾아야 한다. 지금부터 서로 데이터를 교환하는 8개의 처리소자의 예제를 통해 단순한 공유 버스 구조, 수평적인 버스 분할 구조, 그리고 트리구조 기반의 버스 분할 구조의 평균적인 소모 에너지에 대해 살펴본다. 그림 2는 8개의 처리소자간의 연결을 보여준다. 모든 처리소자가 상호 데이터 교환을 할 수 있기 때문에 완전 그래프(complete graph)의 형태로 모델링 하였다. 본 논문에서는  $sw=0.5$ ,  $V_D=1$ 로 정해져 있다고 가정한다.

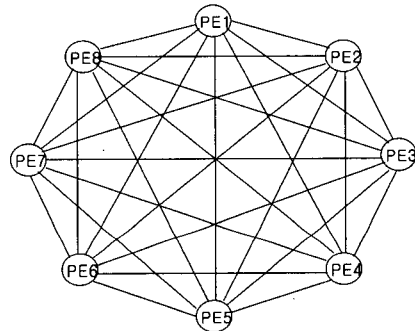


그림 2 연결된 8개의 처리소자

3.1.1 단순한 공유(shared) 버스 구조

단순한 공유 버스 구조에서는 모든 처리소자가 양방향 공유 버스에 연결되어 있다. 임의의 처리소자 사이에 데이터 교환이 이루어지는 동안 신호가 버스 선 전체에 공급된다. 모든 단위길이는  $707 \mu m$ 로 정의하며, 1로 표준화(normalization)한다. 처리소자의 크기는  $2 \times 2$ 로 같고, 버스와 처리소자간의 거리는 1, 이웃한 처리소자간 거리는 2라고 가정해보자. 버스의 길이를  $L_{BUS}$ 라고 할 때,  $C_{BUS}=L_{BUS}$ 로서 버스 선상의 커패시턴스는 버스 선의 길이로 표준화(normalization)한다. 만약  $PE_1$ 에서  $PE_7$ 로 데이터를 전송한다면 그림 4에서 나타난 것과 같이 두 처리소자간 거리는 8이고 버스의 총 커패시턴스는 14가 된다. 임의의 처리소자에서 특정 처리소자로 데이터를 전송할 때마다 버스 전체에 신호 천이가 발생하므로 평균 소모 에너지는  $E_i$ 과 같다.

시스템의 버스를 여러 개의 하위 버스로 묶는 과정은 처리소자간의 데이터 교환 확률이 높은 것끼리 maximum matching 알고리즘[5]을 이용하여 묶을 수 있다.

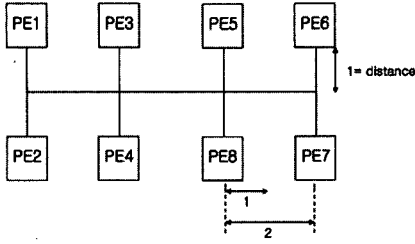


그림 3 단순한 공유 버스 구조

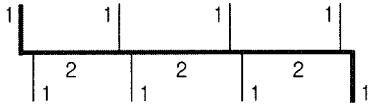


그림 4 그림 3 구조의 PE1과 PE8의 데이터 교환 경로

그림 3은 표 4의 데이터 교환 확률 분포를 가지고 maximum matching 알고리즘을 적용한 것이다.

$$E_1 = 0.5 \times sw \times C_{BUS} \times V_D^2 \quad (3)$$

그림 3의 경우  $C_{BUS}=14$ 이므로  $E_1=3.5$  가 된다.

### 3.1.2 수평적인 2 그룹 버스 분할

수평적 2 그룹으로 분할한 그림 5의 구조에서,  $C_{BUS1}$  과  $C_{BUS2}$ 는 각각  $BUS_1$ 과  $BUS_2$ 의 물리적인 커패시턴스로 정의한다. 그림 5의 두 버퍼의 크기가 처리소자에 비하여 매우 작고 커패시턴스 역시 무시할 만큼 작다고 가정하면  $PE_1$ 과  $PE_7$ 의 데이터 교환 경로의 거리와 커패시턴스는 앞의 3.1.1절에서와 같은 방법으로 계산할 수 있다.

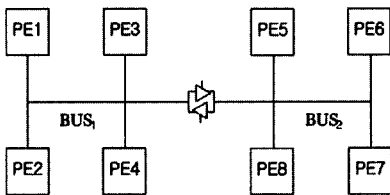


그림 5 수평적 2 그룹 버스 분할

그림 5와 같이 한 쌍의 버퍼로 분할된 버스 구조에서  $BUS_1$ 과  $BUS_2$ 는 분할된 두 버스이다. 예를 들어, 같은 그룹 내의  $PE_1$ 과  $PE_4$ 이 데이터 교환을 한다면, 데이터 교환 경로의 거리는 4이고 커패시턴스는 7이다.

$$P_{BUSx} = \sum_{i \in BUSx} \sum_{j \in BUSx, i \neq j} exchange(P_i, P_j), (k=1, 2) \quad (4)$$

$P_{remaining}$  : data exchange probability except  $P_{BUS1}$  and  $P_{BUS2}$ . (5)

일 때, 2 그룹의 수평적인 버스 분할 구조의 평균적인 에너지 소비  $E_2$ 는

$$E_2 = 0.5 \times sw \times V_D^2 [ C_{BUS1} P_{BUS1} + C_{BUS2} P_{BUS2} + (C_{BUS1} + C_{BUS2}) P_{remaining} ] \quad (6)$$

이다. 8개의 처리소자가 표 4의 데이터 교환 확률 분포를 갖는다면  $E_2=2.16475$ 이다. 이 값은 표 4의 경우에 maximum matching 알고리즘을 적용하여  $\{PE_1, PE_2, PE_3, PE_4\}$ 과  $\{PE_5, PE_6, PE_7, PE_8\}$ 이 각각 같은 로컬 버스에 포함되도록, 버스가 2 그룹으로 분할된 경우에 해당한다.

### 3.1.3 수평적인 4 그룹 버스 분할

본 논문에서 제안하는 트리구조 기반의 버스 분할 방법과 수평적인 버스 분할 방법의 보다 정확한 비교를 위해 4 그룹의 수평적인 버스 분할 구조에 대해 살펴본다.

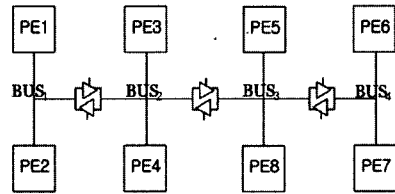


그림 6 수평적인 4 그룹 버스 분할

표 4의 경우에 maximum matching 알고리즘을 적용하면 처리소자들을 4개의 cluster로 묶을 수 있다.

Cluster 1 :  $\{PE_1, PE_2\}$

Cluster 2 :  $\{PE_3, PE_4\}$

Cluster 3 :  $\{PE_5, PE_8\}$

Cluster 4 :  $\{PE_6, PE_7\}$

$C_{BUSi}$ 는  $BUS_i$  ( $i=1, 2, 3, 4$ )의 물리적인 커패시턴스를 의미하고  $P_{BUSi}$ 는 식 (4)과 같다.  $C_{path(i,j)}$ 는  $BUS_i$ 에서  $BUS_j$ 로 데이터 전송이 발생할 때 해당 경로의 커패시턴스를 나타내고,  $P_{(i,j)}$ 는  $BUS_i$ 의 처리소자와  $BUS_j$ 의 처리소자 사이에 데이터 교환이 발생할 확률을 의미한다. 예를 들어, 그림 6의 구조가 표 4의 데이터 교환 확률 분포를 가질 때

$$\begin{aligned} P_{(2,3)} &= \sum_{i \in BUS2} \sum_{j \in BUS3} exchange(PE_i, PE_j) \\ &= 0.019 + 0.014 + 0.002 + 0.012 \\ &= 0.047 \end{aligned}$$

이다. 전체  $m$ 개의 로컬 버스가 존재한다면 4 그룹의 수평적인 버스 분할 구조의 평균적인 에너지 소모는  $E_3$ 과 같다.

$$E_3 = 0.5 \times sw \times V_D^2 \quad (7)$$

$$\left[ \sum_{i=1}^m C_{BUSi} \cdot P_{BUSi} + \sum_{i=1}^{m-1} \sum_{j=i+1}^m C_{path(i,j)} \cdot P_{(i,j)} \right]$$

식 (7)의 대괄호 안의 두 번째 항은 데이터 교환이 가능한 모든 로컬 버스의 쌍을 나타낸다. 즉, 그림 6 구

조의 (BUS<sub>1</sub>, BUS<sub>2</sub>), (BUS<sub>1</sub>, BUS<sub>3</sub>), (BUS<sub>1</sub>, BUS<sub>4</sub>), (BUS<sub>2</sub>, BUS<sub>3</sub>), (BUS<sub>2</sub>, BUS<sub>4</sub>), (BUS<sub>3</sub>, BUS<sub>4</sub>)를 의미한다. 예를 들어 BUS<sub>1</sub>과 BUS<sub>3</sub>에 연결된 PE간에 통신이 필요하면 BUS<sub>2</sub>를 통과해야하므로  $C_{path(1,3)} = C_{BUS1} + C_{BUS2} + C_{BUS3}$ 이다. 표 4의 경우 그림 6의 경우 소모되는 에너지는  $E_3=1.92725$ 이다.

3.1.4 트리구조 기반의 버스 분할

처리소자가 8개이므로 처리소자들은 4개의 cluster로 묶일 수 있다. 처리소자들은 maximum matching 알고리즘을 이용하여 짝지을 수 있다. cluster로 묶인 처리소자들을 각 로컬 버스에 배치하기 위해서 식 (1)의  $P_{cost}$ 를 계산해야 한다. 그림 7에서 H 형태 트리구조의 왼쪽 (BUS<sub>l,1</sub>, BUS<sub>l,2</sub>)과 오른쪽 (BUS<sub>r,1</sub>, BUS<sub>r,2</sub>)에 cluster들을 배치할 때 3가지 경우를 생각해 볼 수 있다. 각 경우의  $P_{cost}$ 를 비교하기 위해 표 4의 예제를 사용한다.

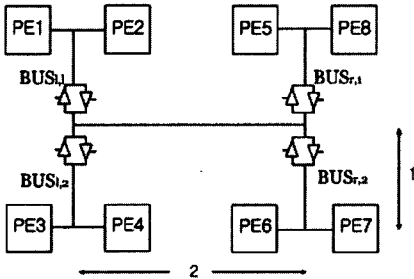


그림 7 4쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 구조

Situation 1 : {Cluster1, Cluster2}, {Cluster3, Cluster4}

$$P_{cost} = 3 \times (0.1 + 0.14 + 0.208 + 0.102) + 8 \times (1 - (0.1 + 0.14 + 0.208 + 0.102)) = 5.25$$

Situation 2 : {Cluster1, Cluster3}, {Cluster2, Cluster4}

$$P_{cost} = 3 \times (0.1 + 0.208 + 0.14 + 0.102) + 8 \times (1 - (0.1 + 0.208 + 0.14 + 0.102)) = 5.25$$

Situation 3 : {Cluster1, Cluster4}, {Cluster2, Cluster3}

$$P_{cost} = 3 \times (0.1 + 0.102 + 0.14 + 0.208) + 8 \times (1 - (0.1 + 0.102 + 0.14 + 0.208)) = 5.25$$

이 예제에서는 그림 7과 같이 4쌍의 버퍼를 삼입하여 트리구조의 버스로 분할할 경우 Situation 1, Situation 2, Situation 3의  $P_{cost}$  값이 동일하다. 그러나 버퍼 쌍을 더 추가하게 되면  $P_{cost}$  값은 줄어들 수 있고 Situation 1, Situation 2, Situation 3의  $P_{cost}$  값도 서

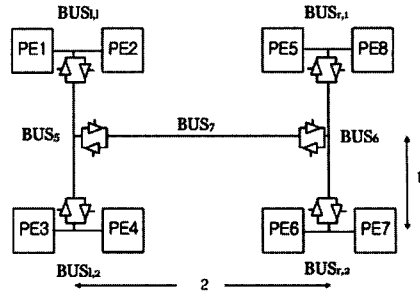


그림 8 6쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 구조

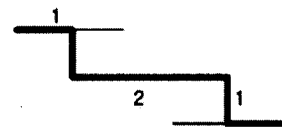


그림 9 그림 7의 PE<sub>1</sub>과 PE<sub>8</sub>의 데이터 교환 경로

로 다를 수 있다.

PE<sub>1</sub>에서 PE<sub>7</sub>로 데이터 전송이 발생하면 그림 9에서 알 수 있듯이 해당 경로의 거리는 6이고 커패시턴스는 8이다. 그림 7의 구조에서 PE<sub>1</sub>로부터 PE<sub>5</sub> 또는 PE<sub>7</sub>로 데이터 전송이 이루어지는 경우와 같이 H 형태 트리구조의 왼쪽에 해당하는 BUS<sub>l,1</sub>, BUS<sub>l,2</sub>와 오른쪽에 해당하는 BUS<sub>r,1</sub>, BUS<sub>r,2</sub> 사이에 데이터 교환이 존재할 때 해당 경로의 전체 커패시턴스를  $C_{mutual}$ 로 표현한다.

따라서 그림 7에서  $C_{BUS1,1} = C_{BUS1,2} = C_{BUSr,1} = C_{BUSr,2} = 3$ 이고  $C_{mutual} = 8$ 이 된다. 그림 7과 같이 4쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 구조에서의 평균 에너지 소모는  $E_4$ 와 같다.

$$E_4 = 0.5 \times sw \times V_D^2 \left[ \sum_{x=1, r} \sum_{y=1, 2} C_{BUSx, y} \cdot P_{BUSx, y} + C_{mutual} \left( 1 - \sum_{x=1, r} \sum_{y=1, 2} P_{BUSx, y} \right) \right] \tag{8}$$

표 4의 예제에서  $E_4 = 1.3125$  값으로 3.1.1절과 3.1.2절, 3.1.3절의 버스 구조 중 가장 낮은 에너지를 소모한다.

그림 8과 같이 6개의 버퍼 쌍으로 버스를 분할하면 그림 7의 구조보다 더 많은 에너지를 절약할 수 있다. 그림 8 구조의 평균 에너지 소모는  $E_5$ 로 나타낼 수 있다.

$$E_5 = 0.5 \cdot sw \cdot V_D^2 \left[ \sum_{x=1, r} \sum_{y=1, 2} C_{BUSx, y} \cdot P_{BUSx, y} + C_{path(l,1, l,2)} \cdot P_{(l,1, l,2)} + C_{path(r,1, r,2)} \cdot P_{(r,1, r,2)} + C_{mutual} \sum_{i=1, 2} \sum_{j=1, 2} P_{(l, i, r, j)} \right] \tag{9}$$

표 4의 예제에서  $E_5 = 1.227$ 로  $E_4$ 보다 적은 에너지를 소비한다. 그러나 버퍼 쌍이 증가하여 시스템의 비용이 증가하고, 시스템의 데이터 교환 확률 분포에 따라서 버

퍼 쌍을 추가하는 것이 오히려 에너지 소비를 증가시킬 가능성도 존재한다. 이와 같은 예가 표 1, 표 2, 표 3의 경우이며, 표 7과 그림 12의 평균 소비 에너지를 통해서 확인할 수 있다.

버스를 분할하면 데이터 교환 확률이 높은 처리소자들을 하나의 로컬 버스로 묶어주기 때문에 처리소자 간에 데이터 전송이 발생할 때 버스 전체에서 신호전이가 발생하는 것보다 평균 에너지 소비를 줄일 수 있다. 이 중에서도 그림 7과 그림 8에 나타난 트리구조로 버스를 분할하게 되면 데이터 통신이 필요한 버스들 사이의 활성화되는 버스의 커패시턴스가 그림 3, 그림 5, 그림 6의 구조보다 작아지기 때문에, 트리구조 기반의 버스 분할 방법이 수평적인 버스 분할 방법에 비해 더 많은 에너지를 감소시킬 수 있다.

3.1.5 버퍼 쌍의 모델링

전력 소모는 버스 선의 길이에 비례하여 증가하므로 정확한 에너지 소비를 계산하기 위해서는 버스 선의 물리적인 커패시턴스를 고려해야 한다[6]. 이 때 분할된 버스들은 그림 10에서와 같이 데이터를 교환하므로 버퍼에서 소모되는 에너지를 함께 고려해야 한다.

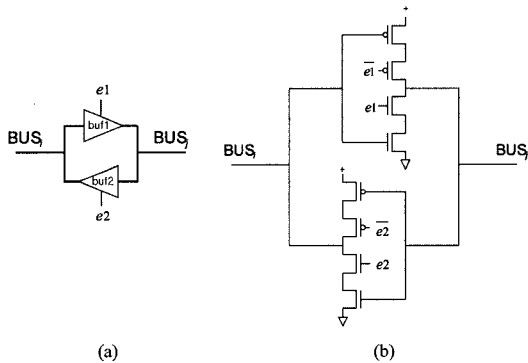


그림 10 (a) 버퍼 쌍에 의한 데이터 교환 (b) 버퍼 모델링

로컬 버스 사이의 데이터 교환은 그림 10의 (a)와 같이 enable 신호에 따라 동작하는 한 쌍의 버퍼를 통해 이루어진다. 버퍼는 일종의 inverter로 생각할 수 있으며 그림 10의 (b)와 같이 버퍼를 clocked inverter로 모델링한다. BUS<sub>i</sub>에서 BUS<sub>j</sub>로 데이터가 전송되는 경우 e1=1, e2=0의 신호가 인가되고, BUS<sub>j</sub>에서 BUS<sub>i</sub>로 데이터가 전송되는 경우 e1=0, e2=1의 신호가 인가된다. 그리고 BUS<sub>i</sub>와 BUS<sub>j</sub> 사이에 데이터 교환이 존재하지 않을 경우에는 e1=e2=0으로 인가되어 두 버퍼 모두 OFF 되도록 한다.

게이트 커패시턴스는 트랜지스터 내부에서 큰 영향을 미치는 요소이다. Poly 게이트와 substrate 사이에 발생

하는 게이트 커패시턴스는 작은 로직 회로내의 커패시티브 부하 (capacitive load)의 대부분을 차지한다[5]. 따라서 inverter의 입력 커패시턴스는 식 (10)과 같이 표현할 수 있다.

$$C_{in} = C_{p\ gate} + C_{n\ gate} \tag{10}$$

버퍼가 ON 될 때 버퍼 커패시턴스를 C<sub>ON</sub>, 버퍼가 OFF 될 때의 버퍼 커패시턴스를 C<sub>OFF</sub>로 표현할 때, C<sub>ON</sub> ≈ C<sub>OFF</sub> 로 가정할 수 있으므로 C<sub>b</sub> = C<sub>ON</sub> = C<sub>OFF</sub> = C<sub>in</sub> 으로 표시한다. 단, 버퍼 커패시턴스는 C<sub>b</sub>로표준화 (normalization)한다.

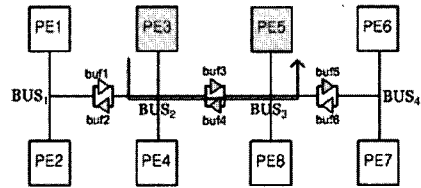


그림 11 PE<sub>3</sub>에서 PE<sub>5</sub>로의 데이터 전송

예를 들어, 그림 6의 4 그룹의 수평적인 버스 분할 구조에서 PE<sub>3</sub>에서 PE<sub>5</sub>로 데이터를 전송한다면 그림 11과 같이 어두운 색으로 표현된 버퍼들의 게이트 커패시턴스가 에너지 소모에 영향을 미치게 되고, buf2의 C<sub>OFF</sub>, buf3의 C<sub>ON</sub>, buf4의 C<sub>OFF</sub>, buf5의 C<sub>OFF</sub>로 인해 C<sub>BUF</sub> = 4C<sub>b</sub>가 된다.

버스 선의 길이를 L<sub>BUSi</sub>, 버스 선의 단위 길이 당 발생하는 커패시턴스를 C<sub>w</sub>라고 했을 때, 버스의 물리적인 커패시턴스 C<sub>BUSi</sub>는 식 (11)으로 표현된다.

$$C_{BUSi} = L_{BUSi} \cdot C_w + C_{BUF} \tag{11}$$

3.2 데이터 교환 확률 분포에 따른 버스 분할 방법

버스를 통해 처리소자 사이에 데이터 통신이 이루어지는 빈도수와 확률은 시스템마다 다를 수 있다. 버스가 분할된 시스템에서 처리소자간의 데이터 교환 확률 분포 특징에 따라 소모 에너지는 변화한다. 따라서 시스템의 데이터 교환 확률 분포에 따라 적절한 버스 분할 방법을 선택해야 버스에서의 소모 에너지 감소 효과를 높일 수 있다. 데이터 교환 확률 분포는 다양한 형태가 있을 수 있다. 여기에서는 다음의 세 가지 경우를 설명한다.

- Case 1 : 전체 처리소자간의 데이터 교환 확률 분포가 거의 균등한 경우
- Case 2 : 시스템 전체에서 데이터 교환 확률이 높은 처리소자가 일부부인 경우
- Case 3 : 처리소자 사이에 데이터 교환 확률 분포가 국부적으로 편중된 경우

버스 분할 방법들을 통해 로컬 버스의 커패시턴스를 감소시킬 수 있고 버스를 분할할수록 로컬 버스의 커패

시턴스는 더 줄어든다. 그리고 서로 다른 로컬 버스의 처리소자들이 데이터 통신을 해야 할 경우에는 해당 경로상의 커패시턴스가 각 로컬 버스의 커패시턴스에 비하여 큰 값을 갖게 되므로 데이터 교환이 국부적으로 특정 처리소자 간에 많이 발생하는 시스템일수록 버스를 여러 개의 로컬 버스로 분할할수록 평균 소모 에너지는 감소하게 된다. 본 논문에서는 4쌍의 버퍼를 포함한 버스 분할 구조와 6쌍의 버퍼를 포함한 버스 분할 구조를 기준으로 어떠한 버스 분할 방법이 더 큰 에너지 감소 효과를 갖는지에 대해 살펴보았다. 단, 6쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 구조인 그림 8을 기준으로 한다.

각각의 데이터 교환 확률 분포에 따라 어떤 버스 분할 방법이 더 효과적인가를 판단하기 위해서  $P_{BUSi}$ 가 전체 데이터 교환 확률 분포에서 어느 정도의 비율을 차지하는가에 대해 고려해야 한다.

이 때,  $C_{BUSi} < C_{BUSi,1} + C_{BUSi,2} < C_{BUS}$  or  $C_{BUSr,1} + C_{BUSr,2} < C_{BUS6} < C_{total}$  이므로  $C_{BUSi}$ 의 데이터 교환확률이 클수록 트리구조 기반의 버스 분할 방법이 더욱 작은 평균 소모 에너지 값을 갖게 된다.

대표적인 예로서 표 1, 표 2, 표 3, 표 4, 표 5, 표 6과 같은 데이터 교환 확률을 갖는 경우는 위에서 설명한 세 가지 데이터 교환 확률 분포에 따라 표 1은 Case 1, 표 2, 표 3 Case 2, 표 4, 표 5, 표 6은 Case 3에 해당한다. 각 경우의 적합한 구조를 선택해보면 대략적으로 다음과 같이 생각해 볼 수 있다.

Case 1 :  $\sum_i P_{BUSi} < (1 - \sum_i P_{BUSi})$  이므로 4쌍의 버퍼를 포함한 트리구조 기반 방법이 대체적으로 적합하다.

Case 2 :  $\sum_i P_{BUSi} < (1 - \sum_i P_{BUSi})$  인 경우 4쌍의 버퍼를 포함한 트리구조 기반 방법이 대체적으로 적합하고,  $\sum_i P_{BUSi} > (1 - \sum_i P_{BUSi})$  인 경우 6쌍의 버퍼를 포함한 트리구조 기반 방법이 대체적으로 적합하다.

Case 3 :  $\sum_i P_{BUSi} > (1 - \sum_i P_{BUSi})$  이므로 6쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 방법이 대체적으로 적합하다.

Case 2의 데이터 교환 확률 분포에서는 국부적으로 데이터 교환 확률이 높은 처리소자간의 데이터 교환 확률이 얼마나 큰 값을 갖느냐에 따라 버스 분할 방법이 변할 수 있다.

4. 실험결과

0.18 $\mu$ m 공정의 5000  $\times$  5000  $\mu$ m<sup>2</sup> 면적의 칩에 전체 8

개의 처리소자가 있는 경우에 대해 살펴본다. 모든 처리소자의 크기는 동일하다고 가정하고 버퍼의 사이즈는 처리소자의 크기에 비해 무시할 수 있을 정도로 작다고 본다. 이러한 조건에서 각각의 버스 분할 방식을 비교하기 위해 표 1, 표 2, 표 3, 표 4, 표 5, 표 6의 처리소자간의 데이터 교환 확률에 대해 실험하였다. 이 실험에서는 [7]의 테크놀로지 파라미터를 사용하였다. Wire의 단위 길이 당 커패시턴스  $C_w = 0.118$  fF, 버퍼의 출력 및 입력 커패시턴스  $C_b = 46.8$  fF, 그리고 Wire의 단위 길이는 0.3 $\mu$ m이다.  $V_D = 1V$ ,  $sw = 0.5$  로 가정한다.

표 7과 그림 12는 표 1, 표 2, 표 3, 표 4, 표 5, 표 6의 데이터 교환 확률 분포에 따른 각 버스분할 방법의 평균 소모 에너지를 나타낸다.  $E_1$ 은 단순한 공유 버스 구조에서의 평균 소모 에너지를 나타내고,  $E_2$ 는 수평적인 2 그룹 버스 분할 구조,  $E_3$ 은 수평적인 4 그룹 버스 분할 구조,  $E_4$ 는 4 쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 구조,  $E_5$ 는 6 쌍의 버퍼를 포함한 트리구조 기반의 버스 분할 구조에서의 평균 소모 에너지를 의미한다. 표 1~6에 주어진 표준편차가 0.00, 0.02, 0.04, 0.06, 0.08, 0.12인 경우의 평균 소모 에너지를 표 7과 그림 12에 표시하였다.

표 7 데이터 교환 확률 분포와 버스 분할 방법에 따른 평균 소모 에너지 (단위는 [f F·V<sup>2</sup>])

표준편차 평균소모에너지	0.00	0.02	0.04	0.06	0.08	0.12
E1	973.3	973.3	973.3	973.3	973.3	973.3
E2	782.4	744.0	678.8	563.1	641.4	507.1
E3	656.0	621.4	501.2	397.8	367.2	268.4
E4	567.9	536.1	481.0	365.0	333.7	227.7
E5	620.7	569.4	534.9	307.8	327.6	166.0

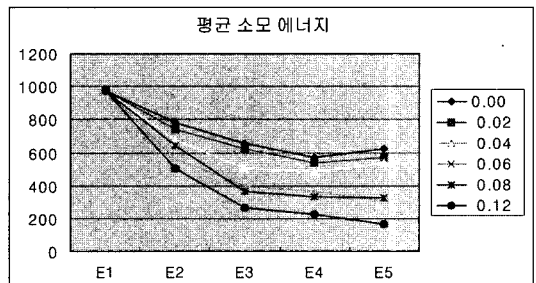


그림 12 데이터 교환 확률 분포와 버스 분할 방법에 따른 평균 소모 에너지 (단위는 [f F·V<sup>2</sup>])

표 7과 그림 12의 결과에 의하면 표준편차가 0.00인 전체 처리소자간의 데이터 교환 확률 분포가 일정한 case1은  $E_4$ 이 가장 적은 값을 갖기 때문에 4쌍의 버퍼

를 포함한 트리구조 기반의 버스 분할구조를 선택하는 것이 가장 적절하다. 그리고 표준편차가 0.02, 0.04인 경우 시스템 전체에서 데이터 교환 확률이 높은 처리소자가 일부뿐일 경우인 case2는 E4가 가장 좋은 결과로서 트리구조 기반의 버스 분할 방식이 가장 적합하다. 따라서 확률 분포가 달라지면 4쌍의 버퍼를 포함한 트리구조 기반의 버스 분할구조가 더 적합한 경우가 생길 수 있다. 마지막으로 표준편차가 0.06, 0.08 0.12인 데이터 교환 확률 분포가 처리소자 사이에 극부적으로 편중되는 case3은 E5의 평균 소모 에너지가 가장 적기 때문에 트리구조 기반의 버스 분할 방법이 가장 적합하다.

소모 전력 면에서는 대부분의 경우 E5가 우수하였다. 그러나 E5는 신호선 당 6 쌍의 버퍼를 사용하므로 버퍼 비용이 크고 표 1, 표 2, 표 3의 경우에는 E4가 E5보다 더 작은 값을 갖는다. 따라서 소모 전력의 감소와 비용의 증가를 비교하여 최적의 구조를 선택할 필요가 있다.

표 8과 그림 13은 TSMC 0.18 $\mu$ m 공정의 5000  $\times$  5000  $\mu$ m<sup>2</sup> 면적의 칩에 버스 분할 방법에 따른 지연시간을 나타낸다.

표 8 버스 분할 방법에 따른 지연시간 (단위 : 초)

E1	E2	E3	E4	E5
1.90E-10	2.00E-10	2.20E-10	1.20E-10	8.00E-11

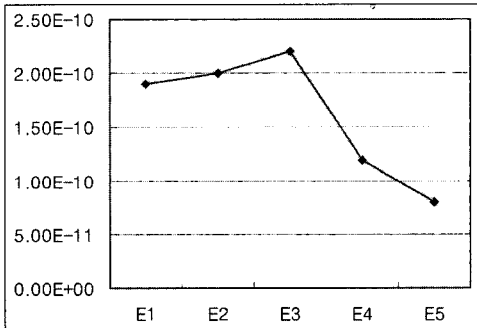


그림 13 버스 분할 방법에 따른 지연시간

E1을 기준으로 할 때, E2, E3의 수평적인 버스 분할 방법은 지연시간이 증가하고, E4, E5의 트리구조 버스 분할방법은 지연시간이 감소하였다. 버스 분할방법에 따른 지연시간의 최대값은 E3의 지연시간  $2.2 \times 10^{-10}$ sec로, E1의 단순한 공유 버스구조의 지연시간인  $1.9 \times 10^{-10}$ sec에 비하여 약간은 증가하였으나 그 증가폭은 크지 않으며, 6쌍의 버퍼를 포함한 트리구조 버스 분할 방법에서는 지연시간이 감소하는 것을 확인할 수 있다. 따라서 적절한 버스의 분할과 버퍼의 사용은 지연 시간을 감소시킬 수 있다.

### 5. 결론

본 논문에서는 시스템의 평균 소모 에너지를 줄이기 위해 새로운 트리구조 기반의 버스 분할 방법과 시스템의 데이터 교환 확률 분포에 따라 어떠한 버스 분할 방법이 적합한지에 대해 연구하였다.

전체 처리소자들 간에 데이터 교환이 거의 비슷한 수준으로 발생하는 경우(Case 1)에 비해 특정 처리소자들 사이에서 데이터 교환이 많이 발생하는 경우(Case 3)의 경우 평균 소비 에너지를 최대 83% 이상 감소시킬 수 있었다. 또한, 버스 분할 방법에 따른 지연시간은 E1을 기준으로 E2, E3의 수평적인 버스 분할방법은 지연시간이 약간 증가하지만, E4, E5의 트리구조 버스 분할구조의 지연시간은 크게 감소하였다. 적절한 버스의 분할과 버퍼의 사용은 소모 전력 감소와 함께 지연시간을 감소시킬 수 있다.

### 참고 문헌

- [1] G. Yeap, "Practical Low Power Digital VLSI Design," Kluwer Academic Publishers, 1998.
- [2] C. Hsieh and M. Pedram, "Architectural energy optimization by bus splitting," IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, No. 4, pp. 408-414, April 2002.
- [3] J. Chen, W. Jone, J. Wang, H. Lu, and T. Chen, "Segmented bus design for low-power systems," IEEE Trans. on Very Large Scale Integration Systems, Vol. 7, No. 1, pp. 25-29, March 1999.
- [4] S. Winegarden, "Bus architecture of a system on a chip with user-configurable system logic," IEEE J. On Solid-State Circuits, Vol. 35, No. 3, pp. 425-433, March 2000.
- [5] S. Even, "Graph Algorithms," Computer Science Press, 1979.
- [6] W. Wolf, "Modern VLSI Design: Systems on Silicon," 2nd ed. Prentice-Hall, 1998.
- [7] Semiconductor Industry Association, National Technology Roadmap for Semiconductor, 1997.



임 호 영

2002년 한양대학교 전자컴퓨터공학부(학사). 2004년 한양대학교 전자전기제어계측공학과(석사). 관심분야는 저전력 설계, 버스분할, 컴퓨터, 반도체





윤 미 선

2005년 대전대학교 전자공학과(학사). 2005년 한양대학교 전자전기제어계측공학과 석사과정. 관심분야는 저전력 설계, H.264, 3D Graphics, 반도체



신 현 철

1978년 서울대학교 전자공학과(학사). 1980년 한국과학기술원 전기및전자공학(석사). 1983년~1987년 U.C. Berkeley Ph.D 1983년~1987년 Fulbright scholarship 1987년~1989년 MTS, AT&T Bell Lab's, Murray Hill N.J., USA. 1989년~현재 한양대학교 전자컴퓨터공학부 교수. 1997년~현재 IDEC 한양대학교 지역센터 센터장. 관심분야는 CAD&VLSI, 통신용 반도체 설계, 저전력 설계



박 성 주

1983년 한양대학교 전자공학과(학사). 1983년~1986년 금성사 소프트웨어 개발 1992년 Univ. of Massachusetts 전기 및 컴퓨터공학과 박사졸업. 1992년~1994년 IBM Microelectronics 연구스텝 1992년~1994년 한양대학교 전자컴퓨터공학부 정교수. 관심분야는 테스트 합성, Built-In Self Test, Scan Design, ATPG, ASIC설계, 고속 신호처리 시스템 설계, 그래프이론