

멀티프로세서 시스템 구성을 위한 시스템 버스의 설계 및 성능평가에 관한 연구

準會員 李 南 宰* 正會員 金 永 川**

A Study on the Design and the Performance Evaluation of System Bus for a MC 68000 Based Multiprocessor System

Nam Jae LEE*, Young Chon KIM** *Regular Members*

要 約 時分割 버스 구조를 갖는 멀티프로세서 시스템의 단점을 보완하기 위하여 각 PE의 로칼메모리를 다른 PE가 직접 액세스하는 것이 가능하도록 시스템 버스를 提案하였다. 이를 위하여 이중 입출 메모리 컨트롤러와 중재기를 설계하였으며 이를 이용한 멀티프로세서 시스템의 具現例를 보였다. 性能평가를 위하여 SPN에 의한 모델링과 부하율에 따른 processing power, 效率 및 시스템 버스의 이용율을 측정하였다.

ABSTRACT In this paper, DPA bus is proposed for implementation of MC 68000 based tightly-coupled multiprocessor system. The DPMC and arbiter are designed that the local memory of each PE can accept memory request both from a local processor and from the system bus.

The performance of the proposed system bus is evaluated by Stochastic Petri Net(SPN) system modeling. The processing power, the efficiency, and the utilization of system bus are simulated for various load factors.

I. 序 論

最近 超集積回路 構成技術의 發達로 인하여

컴퓨터 하드웨어 특히 프로세서 및 메모리등의 價格이 급격히 하락하므로서 저렴한 가격으로 情報處理 및 資料處理를 할 수 있게 되었다. 그러나 現在 많이 사용되고 있는 Von Neumann 구조의 컴퓨터 시스템으로는 速度改善에 한계가 있어 高速 實時間處理를 要求할 경우에는 그의 이용이 제한되고 있는 실정이다.

이에 단일 프로세서의 한계를 넘어 멀티프로세서 시스템의 構成과 사용에 상당한 관심을 갖게

* 全北大學校 電子計算機工學科
Dept. of Computer Eng. Chonbuk Nat'l Univ.

** 全北大學校 電算統計學科
Dept. of Computer Science & Statistics
Chonbuk Nat'l Univ.
論文番號 : 90-11(接受 1989. 6. 8)

되었고 이와 같은 멀티프로세서 시스템은 速度改善과 價格대 性能의 많은 향상 및 信賴性을 增進시키고 있다¹⁾.

멀티프로세서 시스템은 물리적인 연결방법에 따라 時分割 버스 構造, 크로스바 스위치 構造, 멀티포트메모리 構造, 多段相互 接續網構造로 나눌 수 있다. 이들 구조들은 각각의 장, 단점을 갖고있으며 소수의 프로세서를 사용하는 경우에는 時分割 버스 構造가 많이 채택되어진다. 時分割 버스 構造의 멀티프로세서 시스템은 저렴한 가격으로 構成이 가능하고, 소규모 시스템으로부터 필요에 따라 시스템의 擴張 및 修正이 용이하며 具現하기 쉽다는 장점이 있는 반면, 전체 시스템의 性能은 시스템 버스의 效率에 크게 좌우된다.²⁾

이와 같은 시스템 버스는 이용되는 프로세서와 밀접한 관련을 갖으므로 프로세서의 종류에 따라 버스시스템이 개발되어 왔으며, 그 예로는 MOTOROLA 계열의 프로세서를 위한 VMEbus (IEEE 1024)와 인텔계열의 프로세서를 위한 MULTI - II bus 등이 있다.³⁾

VMEbus 등 기존의 standard 버스를 이용하여 멀티프로세서 시스템을 具現할 경우 쉽게 構成이 가능한 반면 PE(Processing Element)간의 통신은 共通 메모리를 통하여만 이루어 지므로 PE간의 통신이 증가할수록 시스템의 性能이 저하된다.

따라서 本 論文에서는 이와 같은 단점을 보완하기 위하여 각 PE의 로칼메모리를 다른 PE가 직접 액세스 하는 것이 가능하도록 시스템 버스를 提案하였다. 이를 위하여 이중 入出이 가능한 메모리 액세스 콘트롤러 (Dual-Port Memory Controller; DPMC)와 時分割 버스 構造에서 핵심이 되는 버스 仲裁機를 설계하였으며, 提案된 버스 시스템을 사용하여 具現한 멀티프로세서 시스템을 SPN(Stochastic Petri Net)에 의한 모델링과 性能評價를 하였다.⁸⁾

II. 시스템 버스의 設計

時分割 버스구조를 갖는 멀티프로세서 시스템은 버스 構造가 시스템 性能 향상에 결정적인 영향을 미칠 수 있으므로 컴퓨터 시스템 디자인은 그 시스템의 應用에 알맞는 버스를 選擇하거나 設計하여야 한다.

멀티프로세서 시스템 構成을 위하여 標準화된 범용 多層構造 버스는 MOTOROLA의 VMEbus, INTEL의 MULTI - II bus, IEEE P 896 bus 등이 있다.

現在 진행되는 標準化 경향에 따라 IEEE-P896 bus가 표면적으로 점점 대두되고 있기는 하나 최근의 컴퓨터 시스템에서 주로 사용되고 있는 버스는 VME bus와 MULTI - II bus이며, 이 2종류의 버스는 어드레스와 데이터의 멀티플렉싱 方式, 同期方式, 버스 仲裁方式, 인터럽트 方式등에서 많은 차이가 있다. 이 2종류 버스간의 차이점은 현재의 추세라면 상대방의 長點 그리고 短點의 보완등으로 거의 비슷한 양상의 버스로 변화될 전망이다. 다만 VME bus는 MC68000 系列을 기초로한 시스템에 적합하고 MULTI - II bus는 INTEL의 80×86에 이상적으로 設計되었다는 점에서 버스의 선택은 시스템에 사용되는 프로세서에 의해 거의 依存하게 될 것이다.

本 論文에서는 각각의 PE는 로칼메모리를 갖고 있으며 이를 다른 PE가 직접 액세스할 수 있도록 設計하여 전체적인 데이터 傳送時間을 줄임으로써 시스템 버스의 사용을 위한 競合을 줄여 시스템의 性能을 향상시킬 수 있도록 하였다. 이를 위하여 DPA(Dual Port Access) 버스를 提案하였으며 MC 68000 系列의 프로세서를 사용하는 멀티프로세서 시스템에 적합하도록 設計하였다.

II-1. DPA버스의 定義

DPA 버스의 構造는 그림1과 같다.

本 버스는 3종류의 信號 그룹과 7종류의 機能 모듈로 構成되어 있다.

데이터 전송 버스(Data Transfer Bus)는 高速 並列 데이터 전송 버스로서 어드레스 버스, 데이

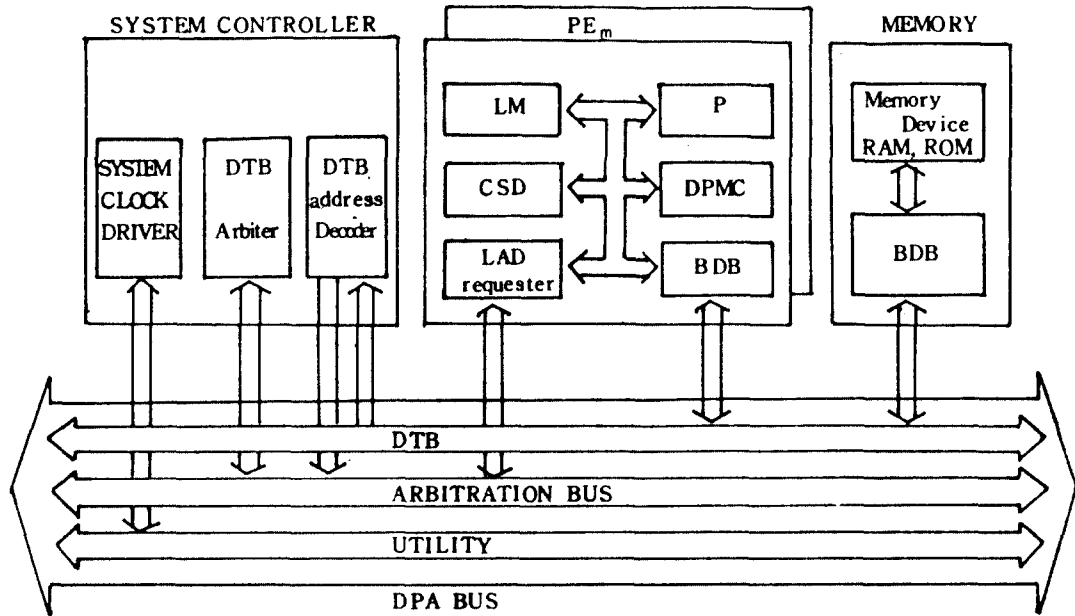


그림1. 시스템 버스 構造
DPA bus structure

타 버스 및 관련 제어 버스로 이루어지며 어드레스 버스는 16, 24bit의 크기를 갖을 수 있고, 데이터 버스는 8, 16bit 크기를 갖도록 하였다. 관련 제어신호로는 \overline{GAS} , \overline{GUDS} , \overline{GLDS} , $\overline{GR}/\overline{GW}$ 등이 있다.

仲裁 버스 (Arbitration Bus)는 다수의 버스 requester가 동시에 버스를 액세스하는 것을 방지하고 共通資源에 대한 사용을 스케줄하기 위한 버스로써 다음과 같은 信號로 構成된다.

- \overline{REQLMi} : PE_i의 로칼 메모리 요구 信號
- \overline{REQCM} : 公同 메모리 요구 信號
- \overline{PiG} : PE_i에게 DTB 사용 허용 信號
- \overline{PiREQ} : PE_i의 DTB 사용 요구 信號

유틸리티(utility) 버스는 버스 시스템의 同期 및 初期化를 위한 信號 그룹으로서 다음과 같은 信號들이 있다.

- \overline{GBERR} : DTB 버스 에러
- \overline{GDTACK} : DTB 데이터 Acknowledge
- \overline{GRESET} : 시스템 리셋
- BCLK : 시스템 버스 클럭(32MHZ)

DTB 仲裁機는 각 PE에게 시스템 버스의

사용을 仲裁하기 위한 것으로 仲裁方式에는 고정 우선순위방식(static priority scheme), 동적우선 순위방식(dynamic priority scheme), 라운드로빈 방식(round-robin scheme)등이 있다. 本論文에서는 소수의 프로세서 시스템에 적합하고 하드웨어의 構成이 용이한 고정우선순위방식으로 設計하였다.

二重入出 메모리 컨트롤러(DPMC : Dual Port Memory Controller)는 로칼 메모리가 속한 프로세서 및 다른 PE의 프로세서에 의하여 로칼 메모리의 직접 액세스가 가능하도록 하기 위한 機能 모듈이다.

로칼 어드레스 디코더(LAD)는 각 PE에서 發生한 어드레스에 의하여 로칼 액세스인지 外部 액세스를 할 것인지 구분하여, 로칼 액세스인 경우에는 CSD(Chip Select Decoder)에 의하여 로칼 메모리를 액세스 한다. 外部 액세스인 경우에는 DTB仲裁機에게 DTB를 사용하기 위한 요구를 한다. DTB 어드레스 디코더는 DTB 어드레스 내용에 따라 어느 PE에 속한 로칼 메모리를 액세스한 것인가를 결정하기 위한 모듈

이다.

양방향 버퍼(BDB : Bidirectional Buffer)는 어드레스와 데이터의 方向을 결정하기 위한 버퍼 모듈이다.

II-2. DPA 버스의 데이터 傳送

II-2.1. 外部 메모리 액세스

共通 메모리 및 다른 PE의 로칼 메모리를 액세스하는 順序는 다음과 같다. PE가 어드레스를 발생하면 로칼 어드레스 디코더에 의하여 버스 仲裁機에게 시스템 버스 使用要求信號(\overline{PiREQ})를 발생한다. 이때 버스 仲裁機는 仲裁 알고리즘에 따라 시스템 버스 許容信號(\overline{PiG})를 발생하며 이에 의하여 BDB가 시스템 버스의 使用을 위하여 選擇되어진다. 따라서 시스템 버스를 통하여 어드레스 및 데이터의 傳送이 가능해진다. 이에 대한 타이밍圖는 그림2와 같다.

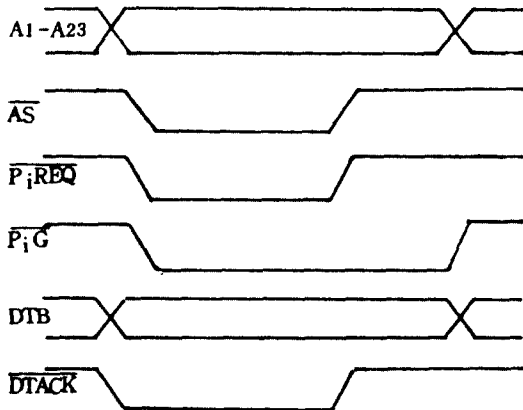


그림2. 外部 메모리 액세스 타이밍 圖
Timing diagram of External memory access

II-2.2. 内部 메모리 액세스

다른 PE로부터 로칼 메모리 액세스가 이루어지는 順序는 다음과 같다. DTB 어드레스 디코더는 수신된 어드레스에 의하여 \overline{REQLMi} 信號를 발생시키며 이 信號가 DPMC에 가해져 BR 信號를 발생한다. 이때 MPU는 현재의 버스 사이클

을 끝내고 \overline{BG} 信號를 액티브 상태로 하며 이 信號에 의하여 DPMC는 \overline{BGACK} 信號를 액티브 상태로 하며 BDB는 内部 메모리 액세스 상태로 活性化된다. 타이밍 圖는 그림 3과 같다.

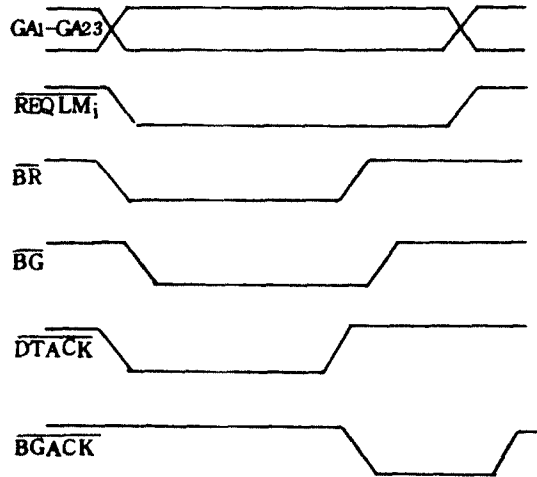


그림3. 内部 메모리 액세스 타이밍 圖
Timing diagram of internal memory access

III. 멀티프로세서 시스템의 具現例

DPA 버스를 이용하여 최대 4개의 PE를 갖는 멀티프로세서 시스템을 設計하였다. 각 PE는 256Kbyte까지의 로칼 메모리를 갖으며 시스템의 전체적인 메모리 배열은 그림 4와 같다.

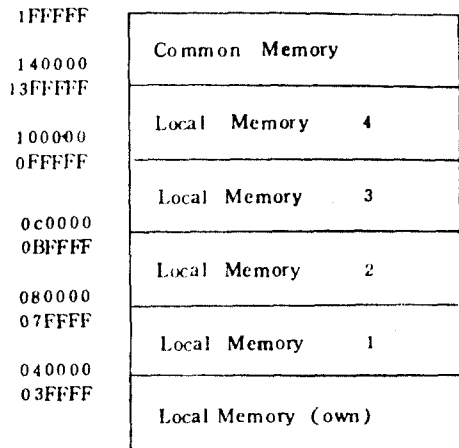


그림4. 멀티프로세서 시스템의 메모리 배열
Memory map of Multiprocessor system

각각의 PE는 자신의 로칼 메모리에 고유 어드레스(000000~03FFFF)를 할당하였으며 外部에서의 액세스를 위하여 상대 어드레스를 할당하여 外部액세스와 로칼 액세스를 구분할수 있도록 하였다. 메모리 배열에서 알 수 있듯이 PE_i은 로칼 액세스인 경우에는 000000~03FFFF이나 다른 PE가 PE_i의 로칼 메모리를 액세스할 경우에는 040000~07FFFF이다.

그림 4의 메모리 배열에 의하여 LAD의 出力信號는 다음과 같다.

$$\begin{aligned} \overline{LMREQ} &= A23 + A22 + A21 + A20 + A19 + A18 \\ &\quad + \overline{AS}(1) \\ \overline{PiREQ} &= \overline{A23 + A22 + A21 + A20 + A19 + A18 + AS} \\ &= (\overline{LMREQ})' \end{aligned} \quad (2)$$

DTB 어드레스 디코더의 出力信號는 다음과 같다.

$$\overline{MREQ} = GA23 + GA22 + GA21 \quad (3)$$

$$\overline{LM1REQ} = \frac{(\overline{MREQ})' \cdot (GA20)' \cdot (GA19)'}{(GA18) \cdot \overline{GAS}} \quad (4)$$

$$\overline{LM2REQ} = \frac{(\overline{MREQ})' \cdot (GA20)' \cdot (GA19)'}{(GA18) \cdot \overline{GAS}} \quad (5)$$

$$\overline{LM3REQ} = \frac{(\overline{MREQ})' \cdot (GA20)' \cdot (GA19)'}{(GA18) \cdot \overline{GAS}} \quad (6)$$

$$\overline{LM4REQ} = \frac{(\overline{MREQ})' \cdot (GA20)' \cdot (GA19)'}{(GA18) \cdot \overline{GAS}} \quad (7)$$

$$\overline{CMREQ} = \frac{(\overline{MREQ})' \cdot (GA20)' \cdot (GA19)'}{(GA18) \cdot \overline{GAS}} \quad (8)$$

메모리 칩은 32kbyte의 용량을 갖는 static RAM (62256)을 사용하였다.

III-1. DPMC의 設計

DTB버스로부터 로칼 메모리의 요구가 있으면 이 信號에 의한 로칼 메모리 액세스가 許容되기까지의 DPMC의 狀態遷移圖는 그림 5와 같다.

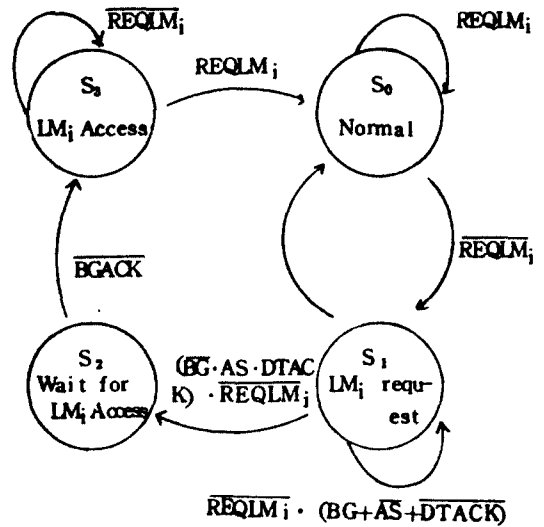


그림5. DPMC의 狀態 천이도
State transition diagram of DPMC

상태 천이도로부터 D Flip-Flop을 이용하여 구현하기 위한 제어표는 표1과 같으며 Q₁은 \overline{BR} , Q₂는 \overline{BGACK} 와 \overline{DTBEN} 信號이다.

표1. DPMC의 제어표
Control table of DPMC

state	Previous Sstate		Next state		Control	
	Q ₁	Q ₂	Q ₁	Q ₂	D ₁	D ₂
S ₀	1	1	\overline{REQLMi}	1	\overline{REQLMi}	1
S ₁	0	1	\overline{REQLMi}	$\overline{AS \cdot DTACK \cdot BG \cdot REQLMi}$	\overline{REQLMi}	$\overline{AS \cdot DTACK \cdot BG \cdot REQLMi}$
S ₂	0	0	1	0	1	0
S ₃	1	0	1	\overline{REQLMi}	1	\overline{REQLMi}

$$D_1 = (\overline{REQLMi})' \cdot Q_2 \cdot \overline{RESET} \quad (9)$$

$$D_2 = \overline{Q_1} \cdot \overline{Q_2} \cdot \overline{RESET} + Q_1 \cdot \overline{Q_2} \cdot (\overline{REQLMi}) +$$

$$\frac{\overline{RESET} + \overline{Q_1} \cdot \overline{Q_2} \cdot (AS)' \cdot (\overline{DTACK})' \cdot \overline{BG}}{\overline{REQLMi} \cdot \overline{RESET}} \quad (10)$$

이다.

III-2. DTB 仲裁機의 設計

고정 우선순위방식을 적용하여 設計하였다. 우선순위는 PE1 이 가장 높고, PE4가 가장 낮다. SREQ는 하드웨어 개발 시스템인 VME / 10이 시스템버스의 사용을 위한 요구이며 PE1 보다 우선순위가 높도록 하여 멀티프로세서 시스템과 정보전송이 가능하도록 하였다. 이에 대한 狀態遷移圖는 그림 6과 같다.

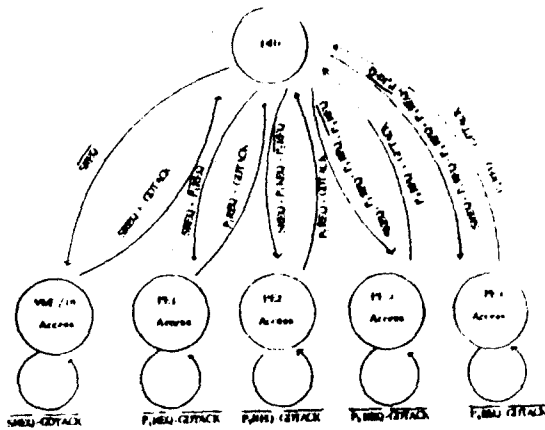


그림6. DTB 仲裁機의 狀態 遷移圖
State transition diagram of DTB arbiter

IV. SPN에 의한 모델링과 性能評價

M · K Molloy가 提案한 Stochastic Petri Net (SPN)을 이용하여 시스템의 모델링 및 性能評價를 하였다. SPN은 일반적인 Petri Net에 性能分析을 위하여 각 천이에 관련된 유한집합 G를 추가하여 다음과 같이 定義한다.⁹⁾

$$SPN \triangleq (P, T, A, M, G)$$

여기서 $P = (p_1, p_2, \dots, p_n)$

$$T = (t_1, t_2, \dots, t_m)$$

$$A \supset (P * T) \cup (T * P)$$

$$M = (m_1, m_2, \dots, m_n)$$

$$G = (g_1, g_2, \dots, g_m)$$

DPA 버스를 사용하여 具現한 멀티프로세서 시스템을 모델링하기 위하여 프로세서의 狀態를 다음과 같이 定義하였다.

- 1) Active : 프로세서가 자신의 로칼 메모리를 사용할 때
 - 2) Accessing : 프로세서가 DTB를 사용할 때
 - 3) Waiting : 프로세서가 DTB를 사용하기 위하여 기다릴 때
 - 4) Blocking : 외부 액세스에 의하여 자신의 로칼 메모리 사용이 정지된 때
- DPA 버스를 사용하여 2대의 프로세서로 具現한 시스템의 모델링은 그림 7과 같다.

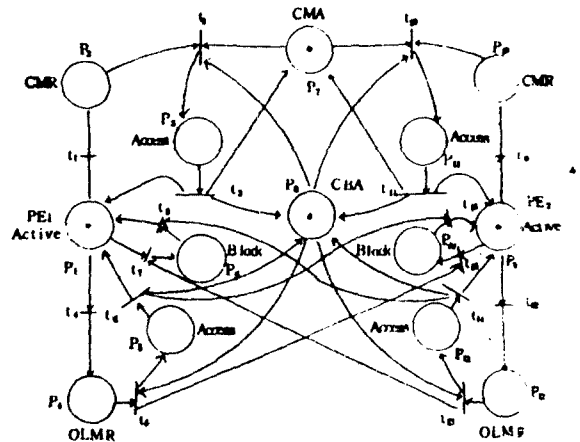


그림7. 2대의 PE를 갖는 시스템의 모델링
SPN model of system with two processors

처음에 모든 프로세서는 자신의 로칼 메모리 액세스 상태에 있으며, 外部 로칼 메모리 액세스를 요구할 때 OLMR(other local memory request) 상태가 되며, 이때 DTB버스가 사용 가능하면 액세스 상태가 된다. 공통 메모리 액세스를 요구할 때 프로세서는 CMR(common memory request) 상태가 된다. 여기서 DTB가 사용 가능

하고 공통 메모리가 액세스 가능하면 Accessing 상태가 된다. 그렇지 않으면 CMR 상태에서 대기하게 된다. 이와 같은 시스템의 상태 변화는 토큰을 運用하므로써 이루어지며 그림 7의 토큰 분포는 初期狀態를 나타낸다.

각 천이에 관련된 점화율은 다음과 같이 定義한다.

$$g_1 = g_4 = g_7 = g_9 = g_{12} = g_{15} = \lambda / 2 \quad (12)$$

$$g_3 = g_6 = g_8 = g_{11} = g_{14} = g_{16} = \mu \quad (13)$$

$$g_2 = g_5 = g_{10} = g_{13} = \alpha \quad (14)$$

λ 는 로칼 액세스率로서 액티브 상태에 있다가 DTB버스의 사용을 요구하는 율(rate)로서 CMR 상태와 OLMR상태가 교대로 발생한다고 간주하였다. μ 는 外部 액세스율로서 DTB사용 후에 로칼 메모리 액세스 상태로 돌아오는 率이다. α 는 버스 仲裁와 관련이 있으며 프로세서가 액티브상태 및 액세스 상태에 있는 시간보다 버스 仲裁와 解際에 걸리는 시간이 매우 작으므로 무시할 수 있다.

4대의 프로세서를 사용하는 경우는 그림 2의 확장으로 볼 수 있으며 같은 方法으로 모델링할 수 있다. 모듈화하여 그림 8에 나타내었다.

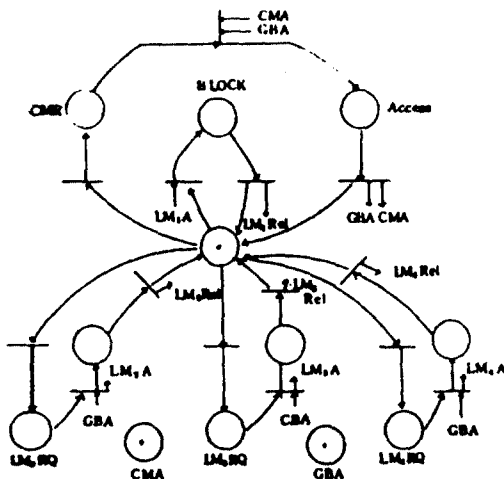


그림8. 4대의 PE를 갖는 시스템의 모델링 SPN model of system with four processors

시뮬레이션을 위하여 다음과 같이 假定한다.

1) 프로세서는 액티브 상태후에 공통 메모리와 다른 로칼 메모리를 교대로 요구한다.

2) 각 프로세서의 로칼 액세스율과 외부 액세스율은 서로 무관한 指數分布函數의 變數이다.

3) 프로세서가 공통메모리 使用을 요구했을 때 버스가 사용 가능하고 공통메모리가 액세스 가능하면 즉각적으로 path가 설정된다. (지연 없음)

4) 프로세서가 外部 로칼 메모리 사용을 요구했을때 DTB가 사용 가능하면 즉각적으로 path가 설정된다.

5) path가 설정될 수 없으면 프로세서가 대기 상태에 있다.

6) DTB의 사용이 완료되면 즉시 해제된다.

시뮬레이션을 위한 入力變數는 平均 로칼메모리 액세스 시간과 平均 外部 액세스 시간이며 이를 이용하여 시스템 負荷率(system load factor) ρ 를 다음과 같이 定義한다.

$$\rho = \frac{\text{平均 공통메모리 액세스시간}(1/\mu)}{\text{평균 로칼메모리 액세스시간}(1/\lambda)} = \frac{\lambda}{\mu} \quad (15)$$

따라서 ρ 는 λ 와 μ 에 의하여 결정된다. 이에 따른 processing power p 는 이용되는 프로세서 갯수의 평균치로서 다음과 같이 定義된다.

$$P = \sum_{n=1}^N \frac{T_n - W_n - B_n}{T_n} \quad (16)$$

여기서 T_n 은 프로세서 n 이 처리한 총시간, W_n 은 프로세서 n 이 대기상태에 있는 총시간, B_n 은 프로세서 n 의 Block상태의 총시간, N 은 프로세서의 數이다.

결과를 확인하기 위하여 2대의 프로세서를 사용하는 경우에는 Markov Chain의 상태 천이도에 의한 解析的인 方法으로도 구하였다.

표2. 부하율에 따른 processing power
Processing power of multiprocessor system

부하율 PE수 알고리즘	2 대		4 대
	MC	SPN	SPN
0.01	1.995	1.999	3.960
0.05	1.976	1.975	3.812
0.10	1.952	1.949	3.646
0.20	1.909	1.907	3.368
0.40	1.833	1.828	2.970
0.80	1.714	1.710	2.512
1.00	1.667	1.663	2.370

식(17)과 같이 2N-tuple 에 의하여 시스템의 狀態를 定義할 수 있다.

$$(m_1, s_1 : \dots : m_i, s_i \dots m_n, s_n) \quad (17)$$

s_i : 는 프로세서 i 의 狀態로서 다음과 같은 값을 갖도록 定義하였다.

- 4 : 내부 로칼메모리 사용
- 3 : 공통 메모리 사용
- 2 : 외부 로칼메모리 사용
- 1 : Block 상태
- 0 : Queue 상태

또한 m_i 는 프로세서 i 가 使用하는 메모리 모듈의 인덱스이며 다음과 같은 값을 갖도록 定義하였다.

- 3 : 공통 메모리
- 2 : 로칼 메모리 2

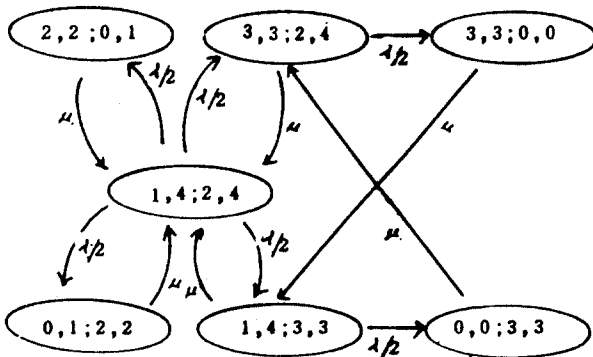


그림9. 그림7과 동가인 MC 상태 천이도
MC state transition diagram for Fig 7.

- 1 : 로칼 메모리 1
- 0 : 비사용

이와 같이 狀態값을 이용하여 2대의 프로세서를 사용하는 경우의 MC 狀態遷移圖를 그림 9에 나타내었다.

그림은 대칭성에 의하여 더욱 간소화할 수 있으며 이에 의한 P는 식(18)과 같다.

$$P = \frac{\rho^2 + 6\rho + 8}{\rho^2 + 4\rho + 4} \quad (18)$$

SPN에 의한 시뮬레이션 결과가 해석적인 方法으로 구한 결과와 거의 같으므로 SPN에 의한 분석 方法의 타당성을 입증하고 있다.

각 프로세서의 효율은 다음과 같이 정의된다.

$$\text{효율}(E) = \frac{\text{Processing Power}}{\text{프로세서의 수}} = \frac{P}{N} \quad (19)$$

표3. 프로세서의 효율 및 시스템 버스의 이용률
Efficiency of each processor and Utilization of system bus

부하율 PE수	효 율		이 용 율	
	2 대	4 대	2 대	4 대
0.01	1.000	0.990	0.013	0.020
0.05	0.988	0.953	0.065	0.095
0.10	0.975	0.912	0.125	0.181
0.20	0.954	0.842	0.235	0.330
0.40	0.914	0.743	0.418	0.554
0.80	0.855	0.628	0.682	0.741
1.00	0.832	0.593	0.750	0.914

그림 10에서 프로세서의 효율이 0.9이상 이기위하여는 2대의 프로세서를 사용할 경우 시스템 부하율이 0.4이하이어야 한다. 이때의 Processing Power는 1.8이상이다. 4대의 프로세서를 사용할 경우 시스템 부하율이 0.1이하

야 하며 Processing Power는 3.6이상이 된다. 버스의 이용율은 시스템 부하율이 증가함에 따라 급격히 증가함을 알 수 있다.

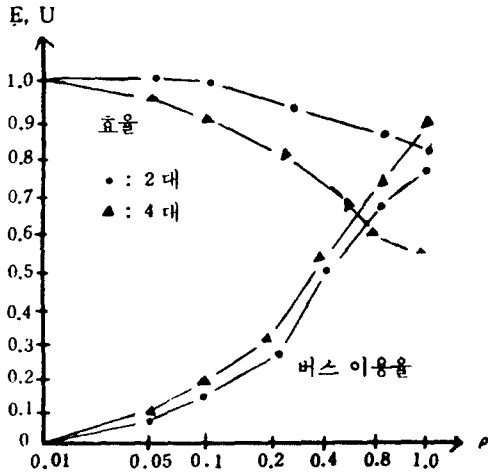


그림 10. 프로세서의 효율 및 시스템 버스의 이용율
Efficiency of each processor and utilization of system bus

V. 結 論

時分割버스 구조를 갖는 멀티프로세서 시스템의 단점을 보완하기 위하여 각각의 PE에 자체의 로칼 메모리를 두었으며 각 프로세서는 다른 프로세서의 로칼메모리를 직접 액세스할 수 있도록 시스템 버스를 設計하였다. 따라서 프로세서간의 通信을 위한 시간과 데이터 전송 시간을 줄임으로써 전체 시스템의 성능을 높일 수 있다.

性能評價를 위하여 SPN에 의한 시뮬레이션에서 시스템 負荷率에 따른 效率率과 Processing Power 및 시스템 버스의 利用率을 측정하였다. 시스템 負荷率이 0.4이하일 경우 Processing Power는 2대, 4대의 프로세서를 사용할 때 각각 1.828, 2.970이상이다. 또한 프로세서 效率率은 2대, 4대의 프로세서를 사용할 때 각각 0.914, 0.743이상이다. 시스템 버스의 利用率은 시스템 負荷率이 0.8이상일 경우 상당히 높아 지므로 시스템 버스의 사용을 위한 프로세서간

의 경쟁이 심각하여 性能 저하의 요인이 됨을 알 수 있다. 따라서 낮은 시스템 負荷率에서는 더욱 높은 性能을 얻을 수 있음을 알 수 있으며 부하율에 따른 적절한 프로세서의 수를 결정하는 것이 가능할 것이다.

本論文은 1988년도 문교부지원 한국학술진흥재단의 자유공모과제 학술연구 조성비에 의하여 연구되었음.

參考文獻

1. J.L. Baer, "Multiprocessing Systems," IEEE Trans. Comp. Vol.C-25, Dec. 1976, pp.1271-1277.
2. M.A. Marsan, G.Balbo and G.Conte, "Comparative performance Analysis of Single Bus Multiprocessor Architectures," IEEE Trans. Comp. Vol.C-31, Dec. 1982, pp.1179-1191.
3. Harvey J.Hindin, "Thirty-two bit system designers face decision time," Computer Design, Feb.1984, pp.27-40.
4. VME bus manufacturers group, "VME bus Specification Manual," Aug.1982.
5. D.Daniel, "Essential Issue Multiprocessor Systems," IEEE Comp. Vol.18, No.6, Jan. 1985, pp.9-27.
6. M.A. Marsan, "Performance models of Multiprocessor Systems," The MIT Press, 1986
7. H.S. Stone, "High-Performance computer Architecture," Addison-Wesley, 1987.
8. C.Lin and D.C. Marinescu, "Stochastic High-Level Petri Nets and Applications," IEEE Trans. Comp., Vol.37, No.7, July, 1988



李 南 宰(Nam Jae LEE) 正會員
1965年 7月18日生
1988年 2月: 全北大學校電子計算機工學
科 卒業(工學士)
1988年 3月~現在: 全北大學校電算統計
學科 大學院 碩士課程中



金 永 川(Young Chon KIM) 正會員
1956年12月10日生
1980年 2月: 高麗大學校 電子工學科 卒
業(工學士)
1987年 2月: 高麗大學校 大學院 電子工
學科 卒業(工學博士)
1987年 3月~現在: 全北大學校電子計算
機工學科 助教授