

실시간 병렬처리를 위한 다중마이크로컴퓨터망의 설계

(Multimicrocomputer Network Design for Real-Time Parallel Processing)

金 鎮 浩*, 高 光 植*, 金 恒 俊*, 崔 興 文*

(Jin Ho Kim, Kwang Sik Koh, Hang Joon Kim, and Heung Moon Choi)

要 約

다중마이크로컴퓨터시스템에서 상호 접속망의 종류 및 그 운영방법에 따라 대기 지연이 고려되도록 network latency time을 재정의 하고, 이를 이용하여 실시간 병렬처리용 다중컴퓨터망의 효율적인 성능평가 및 설계방안을 제시하였다. 정의된 network latency time을 이용하여 망의 지배적 경과시간 및 망의 근사적 성능을 구하고, 이들이 실시간 제약조건을 만족시키도록 접속망을 설계하였다. 시뮬레이션 결과 제안된 방법에 의하면 시스템의 병렬처리성능의 근사적 사전평가가 용이하고, 따라서 실시간 병렬처리시스템의 설계를 효율적으로 할 수 있음을 확인하였다.

Abstract

We proposed a technique to design a multimicrocomputer system for real-time parallel processing with an interconnection network which has good network latency time. In order to simplify the performance evaluation and the design procedure under the hard real-time constraints we defined network latency time which takes into account the queueing delays of the networks. We designed a dynamic interconnection network following the proposed technique, and the simulation results show that we can easily estimate the multimicrocomputer system's approximate performance using the defined network latency time before the actual design, so this definition can help the efficient design of the real-time parallel processing systems.

I. 서 론

순서처리형컴퓨터의 진산처리능력의 한계를 극복하기 위하여 병렬처리용 다중프로세서시스템이나 다중마이크로컴퓨터시스템을 구성하거나 운영하는 방법에 대한 많

은 연구가 진행되어 왔다.^[1] 특히 외부의 연속적인 데이터 입력에 대하여 이를 제한된 시간 이내에 처리해 주어야 하는 실시간 병렬처리시스템에 대해서도 많이 연구되고 있다.^{[1],[2]}

각 노우드컴퓨터가 전용메모리를 갖고 메세지전송에 의해 프로세서간 통신을 하며, MIMD(multiple instruction multiple data stream) 형식의 병렬처리를 하는 다중컴퓨터시스템은 공유메모리를 갖고 공유변수에 의해 통신을 하는 다중프로세서시스템에 비하여 메모리 경쟁이 적다는 장점을 갖고 있으나, 프로세서간의 통신은 통

*正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)

接受日字: 1989年 4月 2日

(※ 이 논문은 1987년도 문교부 학술연구 조성비에 의하여 연구되었음.)

신링크를 통한 메시지전송에 의존하기 때문에 이로인한 노우드간의 통신부담이 시스템의 성능에 큰 영향을 준다. 이와같은 시스템의 통신부담은 접속망의 형태 및 운영방법에 따라 달라질 수 있다.

고정접속망(fixed interconnection network)을 이용한 다중컴퓨터시스템은 접속망의 위상구조와 처리 될 태스크의 데이터전송 구조가 최대정합을 이룰 때에는 최대 성능을 가질 수 있지만 노우드 수가 많아지면 태스크의 최적할당이 어려울 뿐 아니라 태스크 고유의 데이터 전송구조가 접속망의 위상구조와 크게 다를 경우에는 최적태스크 할당을 해도 통신부담이 적지않아 시스템의 성능이 저하되는 단점을 갖는다.^[25]

고정접속망의 이와같은 문제점을 해결하기 위하여 태스크의 데이터 전송구조에 맞춰 접속망의 위상구조를 재구성할 수 있는 가변접속망(reconfiguration network)의 설계 및 다중처리시스템의 구현에 대한 연구결과가 국내외적으로 많이 발표되고 있다.^[7~11]

동질의 마이크로컴퓨터들을 이용한 다중컴퓨터 시스템에서는 각 노우드의 전산처리능력이 일정하므로 시스템의 성능이 접속망의 통신성능에 따라 좌우된다. 따라서 동질의 다중컴퓨터시스템에서는 접속망에 대한 통신능력의 평가가 그 시스템에 대한 근사적성능의 사전평가로 볼 수 있다. Davis 등^[6]은 실시간 병렬처리를 위해 시스템을 구성하고 있는 접속망의 위상구조를 재구성하는데 소요되는 부담을 최소화하는 방안에 대해 발표하였고 Athas 등^[6]은 접속망이 노우드간의 데이터전송을 처리해 내는데 걸리는 시간 즉, network latency time을 정의하여 접속망의 통신성능을 평가하였다. 그러나, Davis 등의 방식에서는 가변접속망이 재구성해낼 수 있는 망의 구조가 한정되어 있고, Athas의 network latency time에 대한 정의에는 대기지연(queueing delay)이 반영되지 않았으며 가변접속망의 종류에 따른 망의 재구성 부담이 정의에 포함되지 않았다. 따라서 병렬처리시스템의 성능평가를 위해서는 접속망의 종류 및 그 운영방법에 따라 이들 파라미터가 고려되도록 network latency time의 재정의가 요구되며 더 나아가서 이를 이용하여 병렬처리시스템의 효율적인 설계방안도 마련할 필요가 있다.

본 논문에서는 다중마이크로컴퓨터시스템에서 상호접속망의 종류 및 운영방법에 따라 대기지연이 고려되도록 network latency time을 재정의하고, 이를 이용한 실시간 병렬처리용 다중컴퓨터망의 효율적인 성능평가 및 설계방안을 제시하였다. 제시된 설계방안에 따라 실시간처리 제약조건하에서 태스크를 처리해 낼 수 있는 다중마이크로컴퓨터망을 설계하고, 설계된 시스템의 병렬처리 성능을 평가 고찰 하였다.

II. 실시간 다중컴퓨터시스템의 설계

1. 상호접속망

다중컴퓨터 시스템에서는 그림 1과 같이 전용메모리를 갖는 각 노우드컴퓨터들이 접속망을 통하여 메시지전송에 의해 프로세스간 통신(interprocess communication)을 하며 일반적으로 MIMD 형태의 병렬 처리를 수행한다.^[13]

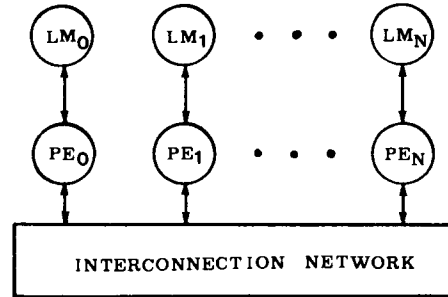


그림 1. 다중컴퓨터시스템의 구조

Fig. 1. Multicomputer system architecture.

따라서 다중컴퓨터시스템에서는 프로세서간의 통신 부담을 줄이는 것이 시스템 성능개선의 관건이며, 노우드 컴퓨터의 처리성능 및 통신링크 대역폭(link bandwidth)이 모두 동일한 동질의 다중컴퓨터 시스템(homogeneous multicomputer system)의 경우에는 병렬처리 태스크가 주어지면 태스크 수행을 위한 노우드간의 통신부담은 결국 접속망의 형태 및 그 운영방법에 따라 달라진다. 따라서 주어진 태스크의 처리에 있어서 가장 효율적인 통신을 할 수 있는 접속망의 선택이 요구된다.

그림 2는 접속망의 종류를 일부 도시한 것이다. 그림 2의 (a) 및 (b)는 고정접속망의 예로서 위상구조가 고정되어 있으므로 접속망의 위상구조와 태스크의 데이터 전송구조가 최대정합(maximum matching)을 이룰 때 최대의 성능이 보장되며 이를 위하여 최적태스크할당 및 최적경로할당 등이 필요하게 된다. 그러나, 노우드의 수 및 부태스크(subtask)의 수가 늘어나면 최적해를 구하기가 어렵고, 특히 접속망의 위상구조와 태스크의 데이터 전송구조가 근본적으로 잘 정합되지 않을 경우에는 시스템의 성능개선에 한계가 따른다.

그림 2의 (c) 및 (d)는 가변접속망의 예로서 이들에서는 태스크의 데이터전송구조에 따라 망의 위상구조를

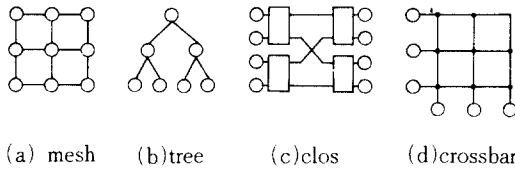


그림 2. 상호접속망의 예

Fig. 2. Examples of interconnection networks.

재구성 시킬 수 있다. 가변접속망의 경우에는 전술한 바와 같이 망의 위상구조를 재구성 하는데 소요되는 부담을 최소화 해야만 최대의 성능이 보장된다. 가변 접속망을 갖는 시스템에서는 분산제어방식으로 위상구조를 재구성 하거나 망제어기(network controller)를 이용하여 망의 위상구조를 중앙제어 하기도 한다.^{[9][10]}

2. Network latency time

다중컴퓨터시스템의 노우드 i에서 노우드 j까지 데이터를 전송하는데 소요되는 시간 즉, network latency time $T_{n_{ij}}$ 는 통신경로 형성시간과 순수한 데이터 전송시간의 합으로 표시될 수 있다. 이 시간은 통신경로 형성 방법에 따라 고정접속망 및 가변접속망의 경우로 각각 분리하여 정의될 수 있다.

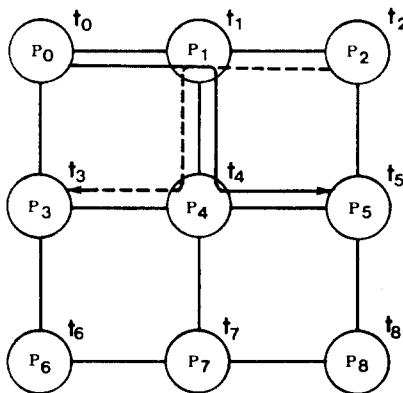


그림 3. 메쉬형 고정접속망에서의 데이터전송 예

Fig. 3. Communication paths in mesh.

그림 3은 태스크 및 경로가 정적할당된 고정접속망에서의 데이터 전송의 한 예를 일부 도시한 것이다. 그림에서와 같이 노우드 P0에서 노우드 P5까지 데이터를 전송하기 위해서는 통신경로상의 노우드 P0, P1 및 P4에

서 라우팅해야 하고, 이 때 노우드 P2에서 P3까지의 전송과는 노우드 P1과 P4 사이의 링크경쟁으로 인한 대기 지연이 발생하게 된다. 따라서, 고정접속망을 이용한 병렬처리 시스템에서의 network latency time $T_{r_{ij}}$ 는 각 노우드에서의 라우팅 알고리즘 수행시간의 합 $T_{n_{ij}}$ 과 데이터의 전송시간 $T_{d_{ij}}$ 의 합으로 표시되는 Athas의 정의식에서 대기지연시간 $T_{q_{ij}}$ 가 추가되어야 한다. 즉,

$$T_{n_{ij}} = T_{r_{ij}} + T_{q_{ij}} + T_{d_{ij}}$$

$$= T_i \times D_{ij} + T_{q_{ij}} + L_{ij}/B_{ij} \quad (1)$$

이다. 여기서 $T_{r_{ij}}$ 는 노우드 i와 노우드 j사이의 각 노우드에서의 라우팅수행 시간 T_i 의 합이고, D_{ij} 는 노우드 i와 j 사이의 거리이다. 또 $T_{q_{ij}}$ 는 노우드 i에서 노우드 j로 데이터를 전송할 때 통신경로상의 링크경쟁으로 인한 대기지연 시간을 표시한다. L_{ij} 는 노우드 i가 노우드 j로 전송해야 할 메시지의 길이를 표시하며, B_{ij} 는 노우드 i와 노우드 j사이의 링크대역폭(link bandwidth)이다. 따라서 network latency time을 최소화하기 위해서는 망의 위상구조와 태스크의 데이터전송구조를 최적화시켜 두 노우드간의 거리 D_{ij} 및 노우드에서의 대기지연 $T_{q_{ij}}$ 를 최소화 해야하고, 각 노우드에서의 라우팅 수행 시간 T_i 를 최소화 해야한다.

한편, 가변 접속망에서는 태스크의 데이터 전송패턴(patten)에 따라 망의 위상구조를 중앙제어 또는 분산제어방식으로 재구성해 줄 수 있다. 그림 4는 태스크의 데이터 전송패턴에 따라 중앙제어방식으로 접속망의 위상구조를 재구성하는 과정을 예시한 것이다. 즉, 어떤 가변접속망이 그림 4(a)와 같은 태스크의 데이터 전송구조를 그림 4(b)와 같이 두번의 접속 패턴으로 만족시킴을 설명한 것이다. 가변접속망의 위상구조를 태스크 t의 데이터 전송구조에 정합시키기 위해 가변접속망이 구성해 내야 할 접속망 패턴의 총 수를 $N_{cmax}(t)$ 라고 하면, 특정태스크 t에 대한 가변접속망의 재구성능도(reconfigurability)는 이 $N_{cmax}(t)$ 에 역비례하며, 따라서 이를 태스크 t에 대한 종속적 재구성능도(task dependent reconfigurability) $R_d(t)$ 라고 정의할 수 있다. 즉,

$$R_d(t) = 1/N_{cmax}(t) \quad (2)$$

이다. 따라서, 가변 접속망을 이용하여 특정 태스크의 데이터 전송구조에 따라 데이터를 전송하기 위해 망제어기가 전송패턴을 형성하여 노우드간 데이터 전송하는 일련의 과정을 반복하므로써 모든 데이터 전송을 마치게 된다. 지금 망제어기가 L_k 개의 링크로 구성된 k 번째 패턴을 형성하는데 소요되는 시간을 T_{patk} 라 하면 이를 다음과 같이 기술 할 수 있다. 즉,

$$T_{patk} = T_{alg} + L_k \times T_{link} \quad (3)$$

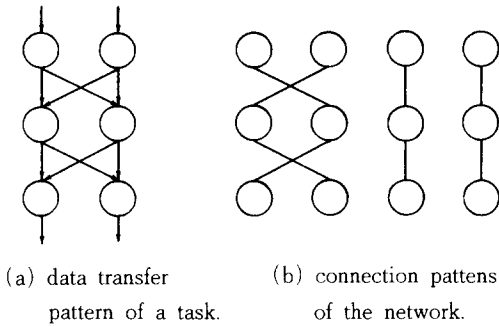


그림 4. 태스크의 데이터전송구조와 그에 따른 가변접속망의 연결패턴

Fig. 4. An example of connection patterns of the dynamic interconnection network to implement the data transfers of a task.

이다. 여기서 T_{alg} 는 망제어기가 패턴형성을 위해 스케줄링(scheduling)알고리즘을 수행하는 시간이고, T_{link} 는 망제어기가 하나의 링크를 설정하는데 소요되는 시간이다. 따라서 중앙제어방식으로 운영되는 가변접속망에서의 network latency time $T_{n_{ij}}^*$ 는 현재 데이터 전송이 포함된 패턴을 형성할 때까지의 대기지연시간 $T_{q_{ij}}^*$ 와 전송패턴 형성시간 $T_{patNc}(t)$ 및 데이터 전송시간 L_{ij}/B_{ij} 의 합으로 다음과 같이 정의할 수 있다. 즉,

$$\begin{aligned}
 T_{n_{ij}}^* &= T_{q_{ij}}^* + T_{patNc}(t) + L_{ij}/B_{ij} \\
 &= \sum_{k=1}^{Nc(t)-1} (T_{patk} + T_{cmax_k}) + T_{patNc}(t) + L_{ij}/B_{ij} \\
 &= (Nc(t) - 1) \times T_{patk} + \sum_{k=1}^{Nc(t)-1} T_{cmax_k} + L_{ij}/B_{ij}
 \end{aligned}
 \tag{4}$$

이다. 식(4)에서 독립된 망제어기가 첫번째 전송패턴을 형성하는 시간은 프로세서 전산처리 시간과 중복(overlap)시킬 수 있으므로 0이 된다. 따라서 식(4)를 보면 $Nc(t)$ 번째 형성된 패턴에서 노우드 i 와 노우드 j 사이의 통신이 일어난다면 network latency time $T_{n_{ij}}^*$ 는 $Nc(t)-1$ 번째 까지의 패턴 형성시간 및 각 패턴에서의 최대 데이터전송시간 T_{cmax_k} , 현재 $Nc(t)$ 번째 패턴형성 시간 및 노우드 i 와 j 사이의 데이터 전송시간을 모두 합한 시간이 됨을 알 수 있다. 따라서, 중앙제어방식 가변접속망으로 구성된 시스템에서 network latency time을 최소화하기 위해서는 망제어기의 패턴형성시간 T_{patk} 를 최소화 하여야 한다.

한편 분산제어방식 가변접속망의 network latency time $T_{n_{ij}}^{**}$ 는 노우드 i 및 j 사이의 링크를 연결하는데

소요되는 시간 $T_{l_{ij}}^*$ 와 노우드 i 가 노우드 j 로 데이터를 전송하고자 할때 발생하는 노우드 경쟁시간 $T_{q_{ij}}^*$ 및 순수데이터전송시간 L_{ij}/B_{ij} 의 합으로 다음과 같이 정의될 수 있다. 즉,

$$T_{n_{ij}}^{**} = T_{l_{ij}}^* + T_{q_{ij}}^* + L_{ij}/B_{ij} \tag{5}$$

이다. 분산제어방식으로 운영되는 접속망에서의 링크설정시간 $T_{l_{ij}}^*$ 는 식(1)에서의 패턴형성시간 T_{patk} 에 비하여 상대적으로 적은 시간이므로 $T_{n_{ij}}^{**}$ 를 최소화 하기위해서는 $T_{q_{ij}}^*$ 를 최소화 할 수 있는 방안을 마련하여야 한다.

3. 실시간 병렬처리

실시간 병렬처리 시스템에서는 처리결과에 논리적인 정확성뿐만 아니라 처리결과를 얻는 데까지 소요되는 허용시간 한계가 정확히 지켜져야 하며, 따라서 시간 제약(timing constraint)과 선행관계(precedence relation) 등의 명확하게 만족되어야 한다. 센서(sensor) 등과 같은 외부의 입력장치로부터 주기적으로 제공되는 데이터를 몇단의 처리를 거쳐 그 결과를 출력하는 실시간 처리시스템은 연속적으로 입력되는 데이터를 거침없이 처리해야 하므로 파이프라인 구조로 모델화 할 수 있다. 그림 5는 파이프라인처리의 예를 도시한 것이다.

파이프라인 처리과정은 그 동기(synchronization) 방식에 따라 크게 두가지 경우로 분리할 수 있다.

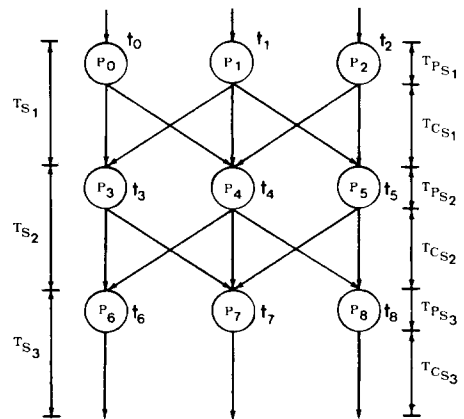


그림 5. 병렬파이프라인처리 및 동기방안

Fig. 5. Parallel pipeline processing and its synchronization.

즉, 파이프라인처리 각 단(stage)의 전산처리와 데이터 전송과정을 합하여 단별로 동기를 시키는 방법이 있고, 또 각 단의 전산처리 과정과 데이터 전송과정을 각각 분

리하여 동기를 시키는 방법이 있을 수 있다.

고정접속망에서는 각 노우드의 전산처리가 끝난 후 라우팅을 행하여 통신경로를 형성해야 데이터를 전송할 수 있어서, 일반적으로 전산처리와 데이터 전송과정을 합하여 동기를 이룰 수 있다. 따라서, 고정접속망의 경우에는 각 단의 경과시간 T_{s_i} 중에서 최대경과 시간인 $\max T_{s_i}$ 를 주기(period)로 하여 동기를 시킨다. 파이프라인 시스템에서 가장 긴 경과시간을 갖는 단의 경과시간을 지배적 경과시간(dominant turnaround time) T_{dom} 라 정의하면, T_{dom} 은 결국 고정접속망을 이용한 파이프라인 시스템에서 파이프라인 처리속도를 결정하게 되며 이는 다음과 같이 계산된다. 즉,

$$T_{dom} = \max_{s_i} T_{s_i}$$

$$T_{s_i} = \max_j (T_{p_i} + T_{n_{ij}})$$

$$= \max_j (T_{p_i} + T_{l_i} \times D_{ij} + T_{q_{ij}} + T_{d_{ij}})$$

(6)

이다. 파이프라인처리 시스템이 실시간 제약조건을 만족하기 위해서는 지배적 경과시간 T_{dom} 이 실시간 데이터 입력 주기 T_{real} 보다는 짧아야 한다. 따라서 실시간 병렬 파이프라인처리 시스템은 다음의 관계식을 만족해야한다. 즉,

$$T_{dom} < T_{real} \tag{7}$$

이다.

따라서 고정접속망으로 구성된 파이프라인처리 시스템의 worst case에서의 속도제고율의 하한을 S_{pmin} 이라 할때 이는

$$S_{pmin} > \frac{T_s}{\max_{s_i} \{ \max_j (T_{p_i} + T_{l_i} \times D_{ij} + T_{q_{ij}} + L_{ij}/B_{ij}) \}}$$

(8)

를 만족시켜야 한다. 여기서 T_s 는 파이프라인처리 시스템에서 수행될 태스크를 순서 처리형 단일 컴퓨터로 처리할 때의 소요시간이다.

한편 중앙제어방식의 가변접속망에서는 망제어기가 태스크의 데이터 전송패턴에 따라 일괄적으로 통신경로를 형성해 주므로 전산처리 과정과 데이터 전송 과정을 분리하여 동기를 시킬 수 있다. 따라서, 각 단별 경과시간 중에서 최대경과 시간에 의해서 성능이 결정되는 고정접속망의 경우와는 달리 최대 전산처리 시간 및 최대 데이터 전송시간의 합에 의해서 그 성능이 결정된다. 따라서 중앙제어방식 가변접속망을 이용한 파이프라인 시스템의 처리속도를 결정하는 지배적 경과시간 T_{dom}^* 는

다음과 같이 된다. 즉,

$$T_{dom}^* = \max_{s_i} T_{p_{s_i}} + \max_{s_i} T_{c_{s_i}}$$

$$T_{p_{s_i}} = \max_j T_{p_i} | s_i$$

$$T_{c_{s_i}} = \max_j T_{n_{ij}}^* | s_i \tag{9}$$

이 되며 여기에 식(4)를 대입하면

$$T_{dom}^* = \max_j T_{p_i} + \max_{i,j \neq i} \{ (N_c(t) - 1) \times T_{pat_k} + \sum_{k=1}^{N_c(t)-1} T_{cmax_k} + L_{ij}/B_{ij} \}$$

(10)

이 되고, 실시간 제약조건은

$$T_{dom}^* < T_{real} \tag{11}$$

이 된다. 따라서, 중앙제어방식 가변접속망을 이용한 파이프라인 처리시스템의 worst case에서의 속도제고율의 하한 S_{pmin}^* 은

$$S_{pmin}^* > \frac{T_s}{\max_j T_{p_i} + \max_{i,j \neq i} \{ (N_c(t) - 1) \times T_{pat_k} + \sum_{k=1}^{N_c(t)-1} T_{cmax_k} + L_{ij}/B_{ij} \}}$$

(12)

이다.

한편 분산제어방식의 가변접속망을 이용한 파이프라인처리 시스템에서의 지배적 경과시간 T_{dom}^{**} 은 다음과 같다. 즉,

$$T_{dom}^{**} = \max_{s_i} T_{s_i} = \max_{s_i} \{ T_{p_i} + T_{n_{ij}}^{**} \}$$

$$= \max_{s_i} \{ \max_{i,j \neq i} (T_{p_i} + T_{l_i}^* + T_{q_{ij}}^* + L_{ij}/B_{ij}) \}$$

(13)

이다. 이 T_{dom}^{**} 도 실시간 처리를 위해서는 다음과 같은 제약조건을 만족해야 한다. 즉,

$$T_{dom}^{**} < T_{real} \tag{14}$$

이다. 따라서 분산제어방식 가변접속망을 이용한 파이프라인처리 시스템의 worst case에서의 속도제고율 하한 S_{pmin}^{**} 은

$$S_{pmin}^{**} > \frac{T_s}{\max_{s_i} \{ \max_{i,j \neq i} (T_{p_i} + T_{l_i}^* + T_{q_{ij}}^* + L_{ij}/B_{ij}) \}}$$

(15)

이 된다. 따라서 태스크와 실시간 제약조건이 주어졌을 때, 이를 처리할 시스템을 구성하기 위해서는 전술한 바와 같은 실시간 제약조건인 식 (7),(11) 또는 (14)를

만족하는 접속망을 설계해야 한다.

고정접속망을 이용한 병렬처리시스템의 실시간 제약 조건식 (7)에서는 태스크와 함께 노우드 컴퓨터 및 링크의 성능이 결정되면 T_{p_i} , L_{ij} , 및 B_{ij} 등은 고정되나, 접속망의 위상구조 및 태스크 할당과 경로할당방법에 따라 T_{1i} , D_{ij} , 및 $T_{q_{ij}}$ 등이 달라진다. 따라서, 실시간처리 시스템을 구성하기 위한 고정접속망을 설계하기 위해서는 식 (7)을 근사적으로 만족할 수 있는 접속망을 선정하고 결국, 라우팅 시간 T_{1i} , 전송거리 D_{ij} 및 대기지연시간 $T_{q_{ij}}$ 등을 최소화 하여 실제 파라미터가 실시간 제약조건식 (7)을 정확히 만족 할 수 있도록 설계하여야 한다.

가변 접속망을 이용한 실시간 병렬처리 시스템의 설계에 있어서도 마찬가지로 지배적 경과시간이 실시간 제약 식(11) 또는 (14)를 근사적으로 만족하도록 접속망을 먼저 선정한다.

실시간 병렬처리 시스템을 위한 접속망의 설계를 위해서 태스크와 함께 노우드 컴퓨터 및 링크의 성능이 결정되면 식 (10)에서는 전산 처리시간 T_{p_i} , 최대 데이터 전송시간 T_{cmax_k} , 메시지 길이 L_{ij} , 링크 대역폭 B_{ij} 등은 결정되나 접속망의 형태 및 운영방법에 따라 패턴 형성수 $N_{cmax}(t)$ 및 패턴 형성 시간 T_{pat_k} 등이 달라질 수 있고 식 (13)에서는 전산 처리 시간 T_{p_i} , 라우팅 시간 T_{1i}^* , 메시지 길이 L_{ij} 및 링크 대역폭 B_{ij} 등은 결정되나 노우드 대기지연 시간 $T_{q_{ij}}^*$ 는 접속망의 운영방법에 따라 달라질 수 있으므로 결국 이 가변파라미터를 조정하여 시스템이 식 (11) 또는 (14)를 정확히 만족할 수 있도록 접속망을 설계하고 운영해야 한다.

Ⅲ. 성능평가 및 고찰

전술한 바와 같은 실시간 병렬파이프라인 처리 시스템의 설계 방안에 따라 다중컴퓨터 시스템을 설계하고 시뮬레이션을 통하여 그 성능을 비교 평가 해 보았다. 또한 8751 단일칩 컴퓨터로 노우드 및 망제어기기를 각각 구성하여 노우드의 전산처리시간, 망제어기의 링크 형성시간 및 데이터 전송시간을 각각 측정함으로써 시뮬레이션 파라미터를 결정하고, 중앙제어 방식 크로스바망을 이용한 실시간처리시스템을 시험설계하여 보았다. 그림 6은 본 논문에서 시뮬레이션 대상으로 설정한 가상적 태스크의 예로서 (a)는 데이터 전송구조가 FFT 구조와 같이 비교적 복잡한 태스크의 예이고 (b)는 데이터 전송구조가 (a)에 비해 비교적 간단한 태스크 구조의 예이다. 그림 6에서 각 노우드의 전산량의 단위는 [killo instruction]이고 데이터 전송량의 단위는 [byte]이다.

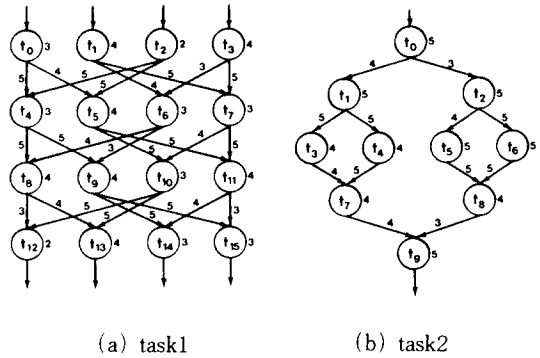


그림 6. 태스크 그래프의 예
Fig. 6. Example task graphs esample.

그림 6(a) 및 (b)와 같은 태스크의 데이터 전송구조에 대한 크로스바 접속망 및 2진 n큐브망의 종속적 재구성능도를 식(2)에 따라 구하면 다음과 같다. 즉,

$$\begin{aligned}
 \text{cross bar} &: Rd(\text{task1}) = 1/N_{cmax}(\text{task1}) = 1/2 = 0.5 \\
 &Rd(\text{task2}) = 1/N_{cmax}(\text{task2}) = 1/2 = 0.5 \\
 \text{binary-n cube} &: Rd(\text{task1}) = 1/N_{cmax}(\text{task1}) = 1/4 = 0.25 \\
 &Rd(\text{task2}) = 1/N_{cmax}(\text{task2}) = 1/2 = 0.5
 \end{aligned}
 \tag{16}$$

이다. 본 연구에서는 1MIPS[million instructions/second]의 전산처리 성능과 17Kbps[killo bits/second]의 데이터 전송 능력을 갖는 단일 칩 컴퓨터를 노우드 컴퓨터 및 망제어 프로세서로 하여 시스템을 설계하였다. 노우드컴퓨터의 성능과 식(16)에서 구한 종속적 재구성능도를 이용하여 식(10)에서 정의한 지배적 경과시간 T_{dom}^* 을 구하면 다음과 같다.

즉, 태스크 1의 경우에는

$$\begin{aligned}
 \text{cross bar} &: T_{dom}^* = 6 + 2 \times T_{pat_k} + 2 \times 4.8 \\
 &= 15.6 + 2 \times T_{pat_k} [\text{msec}] \\
 \text{binary-n cube} &: T_{dom}^* = 6 + 2 \times T_{pat_k} + 4 \times 4.8 \\
 &= 25.2 + 2 \times T_{pat_k} [\text{msec}]
 \end{aligned}
 \tag{17}$$

이 되고, 태스크 2에 대해서는

$$\begin{aligned}
 \text{cross bar} &: T_{dom}^* = 5 + 2 \times T_{pat_k} + 2 \times 2.4 \\
 &= 9.8 + 2 \times T_{pat_k} [\text{msec}] \\
 \text{binary-n cube} &: T_{dom}^* = 5 + 2 \times T_{pat_k} + 2 \times 2.4 \\
 &= 9.8 + 2 \times T_{pat_k} [\text{msec}]
 \end{aligned}$$

이 된다. 태스크 1 및 태스크 2의 가상적 데이터입력 주기를 각각 $T_{real1}=17[msec]$, $T_{real2}=10[msec]$ 로 하였다. 식 (17)은 T_{dom} 를 계산하기 위한 식 (10)에서 T_{pat_k} 만 미지수로 하여 계산한 값이다. 따라서, T_{dom} 를 실시간 제약조건인 식(11)에 대입하면 태스크를 실시간 처리하는데 요구되는 T_{pat_k} 의 상한시간을 계산할 수 있다. 한편, T_{pat_k} 는 식(3)에서와 같이 T_{alg} 및 L_k , T_1 등에 의해서 결정되므로 이들 시간을 최소화하여 시스템이 식(11)을 만족할 수 있도록 크로스바망 및 이를 이용한 다중컴퓨터시스템을 설계하였다.

이때 태스크와 노우드의 수가 주어지면 L_k 도 정해지며, T_{alg} 를 줄이기 위해서는 접속망의 최적패턴 형성순서를 동적으로 결정하지 않고 사전작성하여 망제어기에 기억시키도록 했으며 링크 설정시간 T_1 가 최소가 되도록 링크설정 하드웨어를 최적화 시켰다. 진출한 성능평가 방법에 따라 구한 성능이 식 (11)의 실시간 제약조건을 만족하도록 crossbar 망을 이용한 다중 컴퓨터 시스템을 설계하기 위하여 먼저 T_{pat_k} 가 최소화될 수 있는 가변접속망을 설계하였다. 즉, 16쌍의 단방향 통신 링크들로 각 노우드를 연결하는 스위칭 블록(switching block)과 망 제어기(network controller)를 이용하여 그림 7과 같이 crossbar 망을 구성하였다. 스위칭 블록은 16개의 cell로 구성하였고, 각 셀은 입력과 출력을 연결하는 16×1 MUX와 이를 제어하기 위한 4bit SBCL (switching block control latch)로 구성되어 있어서 crossbar 형태의 접속 구조를 가지고 있다. 망제어기는 노우드 컴퓨터에 memory mapped I/O방식으로 연결된 2개의 PPI(programmable peripheral interface) 및 4×16

decoder로 구성하였다. 스위칭 블록의 제어는 제어기로부터 출력되는 8bit의 SBCL(switching block control data)에 의해서 이루어 지는데, 제어기의 출력 8bit 중에서 상위 4bit는 송신라인을 선택하고 하위 4bit는 수신라인을 선택하도록 하였다. 스위칭 블록 및 제어기의 노우드 컴퓨터들을 접속시킨 시스템의 구성은 그림 8과 같다.

정적인 태스크 할당방식을 전제로 하여 태스크가 주어지면 그 데이터 전송구조에 따라 요구되는 접속망의 연결패턴을 제어기에게 미리 입력시켜 통신링크 설정시간 T_{pat_k} 가 최소화 되도록 함으로써 시스템이 식 (11)의 실시간 제약조건을 만족할 수 있도록 하였다. 연결패턴에 따라 접속망의 위상구조를 재구성하기 위한 망제어기의 동작은 다음 순서를 따르도록 하였다.

- 1) DCSN(dominant communication source node)로 부터 통신링크 설정 요구가 있는지를 확인한다.
- 2) 링크설정 요구가 있을 경우 첫 번째 연결패턴을 형성한 다음 각 노우드들에게 통신링크 형성 완료 신호를 보낸다.
- 3) 현재 설정된 패턴의 DCEN(dominant communication end node)로부터 통신완료 신호가 있는지를 확인하고, 그 다음 연결패턴을 형성한다.
- 4) 데이터 전송구조와 망의 위상구조가 완전정합을 이룰 때까지 $N_{max}(t)$ 개의 패턴에 대해 3)을 반복한 다음 1)부터 다시 수행한다.

한편 진출한 설계방법에 따라 그림 9와 같이 식 (13)을 이용한 분산제어 가변 접속망으로 구성되는 시스템과 식 (6)을 이용한 고정접속망인 환형망으로 구성되는 시스템을 추가로 설계하였다. 분산제어 크로스바망을

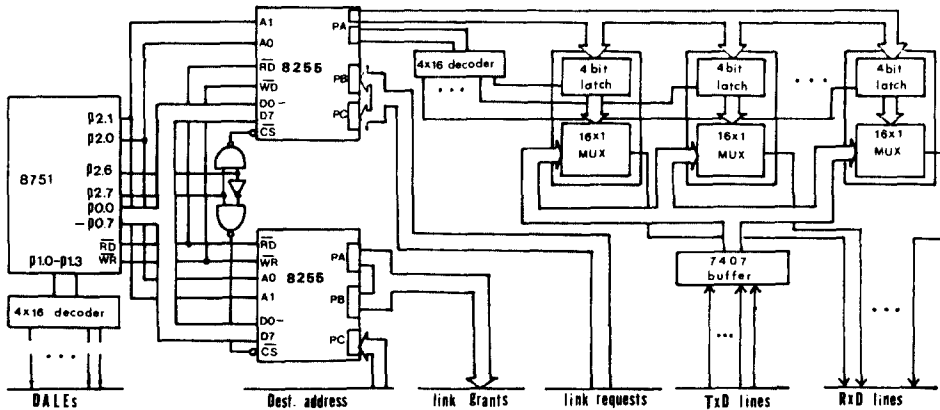


그림 7. 본 연구에서 구성한 가변접속망

Fig. 7. A dynamic interconnection network implemented for this study.

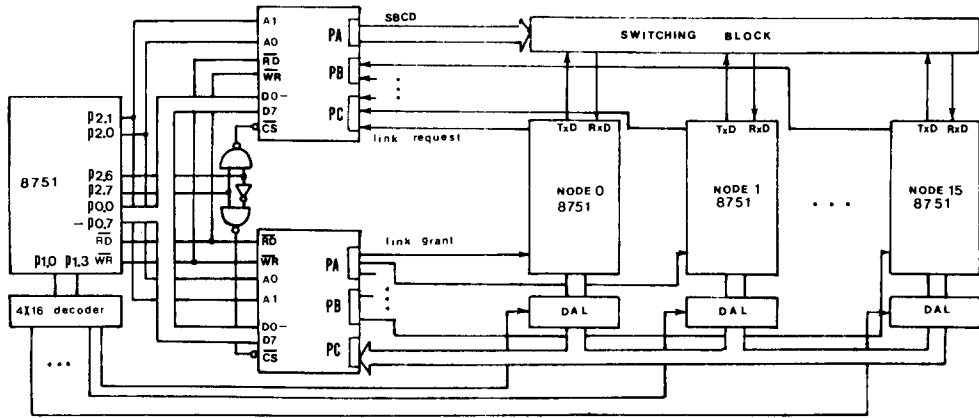


그림 8. 제안된 다중컴퓨터 시스템의 구조
Fig. 8. Proposed multicomputer system architecture.

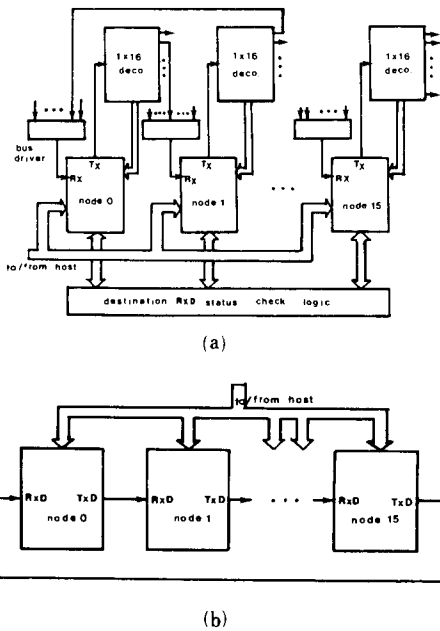


그림 9. 분산제어 크로스바망 (a) 및 환형망 (b)을 이용한 다중마이크로컴퓨터망

Fig. 9. Multimicrocomputer system with a distributed-controlled cross-bar network (a) and ring network (b).

이용한 시스템의 동작은 다음과 같다.

- 1) 호스트컴퓨터는 병렬처리 될 부태스크들을 각 노우드컴퓨터에 할당한다.

- 2) 각 노우드컴퓨터는 각자 할당받은 부태스크를 처리하고 처리한 결과를 다른 노우드컴퓨터로 전송하고자 할때는 다음과 같은 순서로 한다.

- a. 수신측 노우드컴퓨터의 수신 status를 확인한다.
- b. 만일 수신측 노우드컴퓨터의 수신라인이 busy가 아닐 때는 송신 노우드컴퓨터가 link를 설정하고 데이터를 전송한다.
- c. 수신 노우드컴퓨터는 직렬전송 인터럽터에 의해서 status를 busy로 하고 데이터를 수신한다.
- d. 수신 노우드컴퓨터의 데이터수신이 끝난 다음에는 status를 empty로 한다.

- 3) 최종 처리결과는 호스트컴퓨터로 전송한다.

한편, 고속접속망인 환형망으로 구성된 시스템의 동작은 다음과 같다.

- 1) 호스트컴퓨터는 병렬처리될 부태스크들을 각 노우드컴퓨터에 할당한다.

- 2) 각 노우드컴퓨터는 각자 할당받은 부태스크를 처리하고 통신이 필요할 경우는 다음 순서에 따른다.

- a. 최적태스크 및 경로할당이 태스크 수행전에 이루어진 상태이므로 송신노우드는 목적지노우드의 주소와 함께 데이터를 해당경로로 전송한다.
- b. 데이터의 중간 경유 노우드는 할당된 경로에 따라 데이터를 전송한다.
- c. 목적지 노우드컴퓨터는 수신된 데이터를 이용하여 태스크를 처리한다.

- 3) 최종 처리 결과를 호스트로 전송한다.

표 1은 그림 6의 (a) 및 (b)의 태스크들을 그림 8과 같은 중앙제어 크로스바를 이용한 시스템, 그림 9(a)와 같

표 1. 다중마이크로컴퓨터시스템의 성능 평가 비교
Table 1. Performance comparison for the multimicro-computer systems.

interconnection network	Tdom/Sp		task1	task2
	Tdom*	proc.		
centrally controlled cross-bar	[msec]	com.	6	5
		total	9.7	4.9
			15.7	9.9
	Sp		4.59	4.65
distribute-controlled cross-bar	Tdom** [msec]	proc.	6	5
		com.	9.4	4.7
		total	15.4	9.7
	Sp		4.68	4.74
fixed ring network	Tdom [msec]	proc.	6	5
		com.	19.2	4.8
		total	25.2	9.8
	Sp		2.86	4.69

*centrally controlled network.

**distribute controlled network.

은 분산제어 크로스바망을 이용한 시스템 및 그림 9(b)의 환형망을 이용한 시스템에서 각각 수행시켰을 때에 대한 성능평가 결과이다. 태스크 1에 대한 시뮬레이션 결과 지배적 경과시간은 크로스바 접속망을 이용한 시스템이 중앙제어방식으로 운영될 때 Tdom*=15.7 [msec], 분산제어방식으로 운영될 때 Tdom*=25.2 [msec]로서 고정접속환형망을 이용한 시스템의 Tdom=25.2[msec]보다 상대적으로 더 빨리 태스크를 처리할 수 있으며 실시간 제약조건인 식 (11) 및 (14)를 만족함을 알 수 있다. 이는 태스크 1의 데이터 전송구조가 비교적 복잡하여 고정 접속망을 이용할 경우 통신 부담이 커지게되어 실시간 제약조건을 만족하지 못하나, 가변 접속망을 이용할 경우에는 태스크의 데이터 전송구조에 따라 접속망의 위상구조를 가변시켜 통신 부담을 줄일 수 있기 때문이다.

한편 태스크 2를 수행시킨 결과를 보면 환형망으로 구성된 시스템의 속도제고율이 Sp=4.69로서 분산제어방식 크로스바망을 이용한 경우의 Sp=4.74보다는 낮지만 중앙제어 크로스바망을 이용한 경우의 Sp=4.65보다는 비교적 높음을 알 수 있다. 이는 태스크 2의 데이터 전송구조가 환형망의 위상구조와 비교적 잘 정합되기 때문이다. 결국 network latency time을 이용하면 시스템의 근사적 성능의 사전평가가 가능하고, 따라서 실시간 병렬처리시스템의 설계를 보다 효율적으로 할 수 있을

을 확인할 수 있다.

IV. 결 론

다중마이크로컴퓨터시스템에서 상호 접속망의 종류 및 운영방법에 따라 대기 지연이 고려되도록 network latency time을 재정의하고, 이를 이용하여 실시간 병렬 처리시스템의 효율적인 성능평가 및 설계방안을 제시하였다.

제안된 방식에 따라 crossbar 형태의 가변접속망을 갖는 실시간 다중마이크로컴퓨터시스템을 설계하고 시뮬레이션 해본 결과, 가변접속망을 분산제어하는 경우나 중앙제어하는 경우 공히 실시간 제약조건을 만족시킬 수 있음을 확인하였다. 제안된 방식을 이용하면 병렬 처리시스템의 근사적 성능의 사전평가가 가능할 뿐만 아니라 실시간 병렬처리시스템의 설계가 보다 용이함을 알 수 있었다. 앞으로 제안된 설계방식에 따라 확장된 시험 시스템을 구현하고 실험함으로써 좀더 실험적 연구가 더 진행되어야 되리라고 본다.

參 考 文 獻

- [1] L.D. Wittie, "Communication structures for large networks of microcomputers," *IEEE Trans. on Computers*, vol. C-30, pp. 264-273, Apr. 1981.
- [2] H.M. Choi, "An optimal task allocation and the performance evaluations of the multimicrocomputer networks," Ph. D. dissertations, Inha Univ. Aug. 1987.
- [3] G.C. Pathak, "Task division and multimicro-computer system," *Proceeding of the 5th conference on distributed computing system*, Denver, Colorado, pp. 13-17, 1985.
- [4] T.Y. Feng, "Parallel processors and processing," *IEEE Trans. on Computers*, vol. C-26, pp. 97-98, Feb. 1977.
- [5] S.H. Bokhari, "On the mapping problem," *IEEE Trans. on Computers*, vol. C-30, Mar. 1981.
- [6] W.C. Athas, C.L. Seitz, "Multicomputers: Message-Passing concurrent computers," *Computer*, pp. 9-24, Aug. 1988.
- [7] L. Snyder, "Introduction to the configurable highly parallel computer," *Computer*, pp. 47-56, Jan. 1982.
- [8] M. Franklin, "VLSI performance comparison of banyan and crossbar communications networks," *IEEE Trans. on Computers*, vol. C-30, pp. 283-291, Apr. 1981.

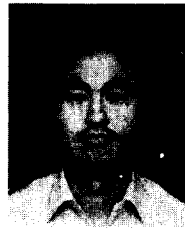
- [9] C. Davis, "Reconfigurable multicomputer networks for very fast real-time applications," National Computer Conference. pp. 491-507, 1982.
- [10] D.J. Kim, G.L. Oh, "The implementation and performance evaluation of multiprocessing system," *Journal of the KIEE*, vol. 25, no. 4, pp. 38-46, 1988.
- [11] S.C. Cheng, "Scheduling algorithms for hard real-time systems," *IEEE Computer*, pp. 150-173, 1988.
- [12] J.A. Stankovic, "Hard-Real time systems," The Computer Society, pp. 1-11, 1988.
- [13] D.I. Moldvan, "Morden parallel processing," Univ. of South California L.A., Jan. 1986.

— 著 者 紹 介 —



金 鎮 浩 (正會員)

1960年 6月 20日生. 1985年 7月 경북대학교 전자공학과 졸업 (B. S). 1988年 2月 경북대학교 대학원 전자공학과 졸업 (M. S) 1988年 3月~현재 경북대학교 전자공학과 박사과정



金 恒 俊 (正會員)

1954年 12月 30日生. 1977年 2月 서울대학교 전자공학과 졸업 1979年 2月 한국과학기술원 전기 및 전자공학과 졸업 (M. S). 1979年 3月~현재 경북대학교 전자계산기공학과 부교수



高 光 植 (正會員)

1957年 2月 11日生. 1980年 2月 경북대학교 전자공학과 졸업 (B. S). 1983年 2月 한국과학기술원 산업전자공학과 졸업 (M.S) 1985年 3月~현재 한국과학기술원 전기 및 전자공학과 박사과정. 1988年 3月~현재 경북대학교 전자공학과 조교수.



崔 興 文 (正會員)

1944年 5月 17日生. 1966年 2月 인하대학교 전기공학과 (전자전공) 졸업 (공학사) 1974년 2월 인하대학교 대학원 전기공학과 (전자전공) 졸업 (공학석사). 1987年 8월 인하대학교 대학원 전자공학과 졸업 (공학박사). 1982年~1983年 뉴욕주립대학교 (SUNY/Stony Brook) 객원교수. 1974年 11월~1989年 9월 현재 경북대학교 공과 대학 전자공학과 교수.