

병렬 컴퓨터의 동향

金明桓, 尹相均

韓國科學技術院 電氣 및 電子工學科

I. 서론

대규모의 문제를 풀기 위한 고성능 컴퓨터에 대한 요구는 구조분석, 기상예측, 석유탐사, 원격탐사, 입자 운동분석, 설계 자동화 및 모의 실험 등 여러 응용 분야에서 계속 증대되어 왔다. 따라서 컴퓨터의 계산 속도를 향상시키려는 연구는 세계 여러 연구소나 회사에서 활발히 진행되어 왔고 앞으로도 계속 많은 투자를 하리라 예상된다. 계산 속도를 향상시키려는 방법으로서 컴퓨터의 중앙 처리 장치(CPU)의 처리 속도를 향상시키기 위한 연구가 계속 진행되어 왔고 많은 성과를 이루었다. CRAY, CYBER 등과 같은 슈퍼 컴퓨터가 고속의 처리 능력을 갖도록 자체 개발된 CPU를 사용한 컴퓨터이다. 몇년 전까지만 하여도 이러한 방식의 컴퓨터가 슈퍼 컴퓨터의 주종을 이루었다. 그러나 이러한 방식의 컴퓨터는 초고속의 전자회로 기술 및 고도의 반도체 제작 기술을 필요로 하며 이에따른 엄청난 비용과 물리적인 문제 때문에 처리 속도의 발전에는 한계가 있으며 그 가격이 너무 비싼 단점이 있다. 한 순간에 하나의 명령어를 처리하는 von Neumann 형의 컴퓨터 구조로는 그 구조의 한계로 인하여 더 높은 성능의 컴퓨터를 만들기가 어렵기 때문에 그 대안으로 여러 개의 프로세서를 사용하여 병렬 처리를 함으로써 고속의 계산 속도를 가능하게 하는 컴퓨터에 대한 연구가 많이 진행되어 왔고 많은 연구 결과들이 제시되었다. 그 결과로서 많은 병렬 컴퓨터들이 연구용으로 제작되었고 상용화되어 사용되고 있는 병렬 컴퓨터도 여러 종류가 있다. 이 방법은 처리속도의 향상에 대한 물리적인 한계를 극복할 수 있고 사용하는 프로세서의 수를 증가 시킴으로써 전체 처리 속도의 급격한 향상이 가능하게 되었다. 최근의 반도체 기술의 급

격한 발전은 이러한 병렬 처리 컴퓨터의 개발과 제조를 더욱 용이하게 하였다. 최근에는 연산 속도가 CRAY-I에 버금가는 상용 프로세서가 나오기에 이르렀고 VLSI 설계, 제조 기술은 병렬 처리 기술에 대한 많은 연구 결과들을 비교적 쉽게 구현 시킬 수 있게 하였다.

앞으로 고속의 계산 능력을 갖는 컴퓨터 시스템의 필요성은 더욱 급속히 증대될 것으로 예상되고 있다. 본 고에서는 병렬 컴퓨터의 여러 종류에 대해 알아보고 최근에 발표된 여러 병렬 컴퓨터에 대한 소개를 하고자 한다. 최근에 개발된 병렬 컴퓨터의 구조와 특징을 알아봄으로써 우리는 병렬 컴퓨터의 개발의 최신 동향 및 앞으로의 연구 방향에 대하여 깊은 이해를 얻을 수 있을 것이다. 본 고의 구성은 다음과 같다. 제2장에서는 병렬 처리와 병렬 컴퓨터의 분류에 대해 살펴 보고, 3장, 4장, 5장에서는 각각 SIMD 병렬 컴퓨터, 공유 메모리 방식 MIMD 병렬 컴퓨터, 메시지 전송 방식 MIMD 병렬 컴퓨터에 대해 최근 연구 사례와 함께 기술한다. 마지막으로 6장 결론과 함께 본 고를 마친다.

II. 병렬 컴퓨터

병렬 처리는 순차(sequential) 처리의 상대되는 개념으로서 프로그램에 내재되어 있는 병렬성을 이용하여 여러 개의 연산을 동일 시간에 처리하는 개념이다. ALU의 carry lookahead 기능, 파이프라인 기능과 계산 기능과 I/O 기능을 분리하여 동시에 처리할 수 있게 하는 것들은 일종의 병렬성을 이용한 것이다. 기존의 슈퍼 컴퓨터에서 많이 사용하는 벡터 프로세서는 병렬 처리를 하는 대표적인 예이다. 벡터 프로세서는 여러 개의

데이터에 대한 동일한 연산을 독립적으로 수행하여 병렬로 처리하게 한다. 벡터 프로세서는 파이프라인 동작을 함께 사용하여 그 성능을 극대화 시킨다. 최근에는 프로세서 내에 여러 개의 처리 장치를 두어서 명령어들 간에 존재하는 병렬성을 이용하여 여러 개의 명령어들을 동시에 처리하도록 하는 슈퍼 스칼라 프로세서도 발표되고 있다. 본 고에서 다루는 병렬 컴퓨터는 이러한 것들과는 구분이 되는 하나의 작업을 여러 개의 프로세서 또는 processing element(PE)를 사용하여 서로 나누어서 필요한 경우 서로 데이터를 교환하면서 협동하여 병렬로 처리할 수 있는 컴퓨터를 말한다.

컴퓨터의 구조는 여러 가지로 분류될 수 있고 대표적인 분류가 Flynn에 의한 분류이다. 1972년 Flynn은 인스트럭션과 데이터 스트림(stream)의 중복도에 따라서 single instruction stream single data(SISD), multiple instruction single data(MISD), single instruction multiple data(SIMD), multiple instruction multiple data(MIMD)의 4가지로 분류하였다. 이들 중에서 여러 개의 처리 장치를 필요로 하는 SIMD와 MIMD 방식의 컴퓨터가 병렬 컴퓨터에 속한다. MIMD 방식의 컴퓨터는 각 프로세서간의 정보 교환 방식과 결합 정도에 따라서 공유 메모리 방식과 메시지 전송 방식으로 나뉠 수 있다. 본 고에서는 SIMD 방식과 공유 메모리 방식, 메시지 전송 방식의 MIMD 병렬 컴퓨터에 대해서 최근에 발표된 상용 컴퓨터를 중심으로 알아보기로 한다. 그림 1은 각 분류에 속하는 대표적인 기종을 나타낸다.

III. SIMD 방식 병렬 컴퓨터

SIMD 컴퓨터는 여러 개의 PE가 각각의 데이터들을 하나의 제어장치의 명령에 따라 동일한 연산을 동시에 수행하는 연산 처리 방식을 가진 컴퓨터를 말한다. SIMD 컴퓨터는 동시 처리하는 방식에 따라 pipelined SIMD vector processor와 병렬 SIMD 컴퓨터로 나누어질 수 있는데, 전자의 방식은 CRAY X-MP, NEC SX, IBM 3090 등 기존의 상용 슈퍼 컴퓨터에서 채택하고 있는 방식으로써 병렬성의 확대에 한계가 있고 그 개발 비용도 너무 많이 소요되는 단점들이 있다. 반면에 병렬 SIMD 컴퓨터는 작은 연산 소자들이 수백에서 수만개 정도가 연결되어 최대의 병렬성을 실현한 구조이다. 본 고에서는 병렬 SIMD 컴퓨터에 대해서 자세히 다루고자 한다.

방 식		기 종
SISD		기존의 단일 프로세서 시스템
MISD		Pipeline processors (not real machines)
SIMD	Pipelined vector processor	CRAY X-MP NEC SX IBM 3090
	병렬 SIMD 컴퓨터	ILLIAC IV IBM GF11 Connection Machine MasPar MP-1 family
MIMD	공유메모리	*미스구조 CMU-Cm Sequent Balance, Symmetry *다단 네트워크 구조 NYU Ultracomputer BBN Butterfly IBM RP3 Alliant Convex
	메시지전송	*하이퍼 큐브 Caltech Cosmic Cube Intel iPSC NCUBE KAIST KAICUBE *메쉬 Ametek Series 2010 Transputer CMP iWarp

그림 1. 컴퓨터의 분류

SIMD 구조를 가진 컴퓨터들로서는 최초의 슈퍼 컴퓨터로 불리는 ILLIAC IV, IBM의 GF11, connection machine, MasPar사의 MP-1 family 등을 들 수 있다. SIMD형 컴퓨터는 동일한 형태의 연산이 무수히 많은 vision, 양자 역학, 유체 역학 및 데이터 베이스등의 특별한 응용에 주로 사용되는 한계가 있지만 VLSI기술의 발전에 힘입어 그 연결 PE의 갯수는 훨씬 증대될 전망이다. 본 장에서는 이들 SIMD 컴퓨터 중 현재 상용화 되어 있는 connection machine과 MP-1 family에 대하여 알아보도록 하겠다.

1. Connection Machine

Connection Machine(CM)은 Thinking Machines Corporation사에서 개발되어 1985년에 처음으로 상품화된 대표적인 SIMD 컴퓨터이다. CM은 4096부터 65536개까지의 processing element(PE)로 구성되어 AI, vision, 유체 역학 및 database등에 광범위하게 이용된다. 각각의 PE는 64Kbit-1Mbit의 local memory를 가진 간단한 1 bit-serial processor이며, 16개의 PE가 daisy chain 형태로써 하나의 CMOS 칩에 집적되어 있다. 또한 전체적인 연결형태는 12 차원의 하이퍼큐브 형태로 되어 있고, 일반적인 SIMD 및 MIMD 컴퓨터와 마찬가지로 VAX나 Sun-4 workstation과 같은 front-end host 컴퓨터에 의해 제어된다. 그림 2는 CM-1의 각 PE의 하드웨어 구조를 보여주고 있다. 그리고, 1987년에 발표된 CM-2는 각 32개의 PE마다 하나씩의 부동 소수점 연산기인 Weitek 3132 칩이 붙어 있어 각각 14 Mflops의 최고 연산 속도를 가지며, 전체 65536개의 PE는 이론적으로 28 Gflops의 최고 연산 속도를 낼 수 있다. CM은 현재 전 세계적으로 50여대가 보급되어 있으며 보급 댓수로서는 미국내에서 Cray Research Inc. 다음 가는 super-computer 회사로 성장하였다.

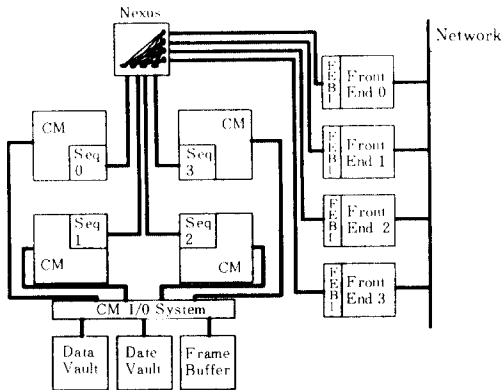


그림 2. Connection machine의 구조

2. MP-1 Family

MP-1 family는 MasPar Computer Corporation에서 개발한 SIMD 구조의 컴퓨터로써, 가장 최신의 MP1216 컴퓨터는 최대 16384개의 PE로 구성되어 있다. CM 컴퓨터와는 달리 각 PE는 1, 8, 16, 32, 64-bits 연산이 가능하며, 32개의 PE가 4x4 grid 형태로 연결되

어 하나의 CMOS 칩에 집적되어 있다. 또한 각 PE cluster는 X-net mesh라고 하는 특별한 형태의 연결 구조를 갖고 있어 이웃하는 8개의 PE cluster와 직접 통신이 가능하다. 그림 3은 X-net mesh의 연결 모양을 나타내고 있다. MP1216 모델의 연산 속도는 평균 1.3 Gflops이다. MP-1 family는 특히 DEC의 VAX-station을 host로 가지며 UNIX 환경에서의 작업을 가능하게 해 주고 있다.

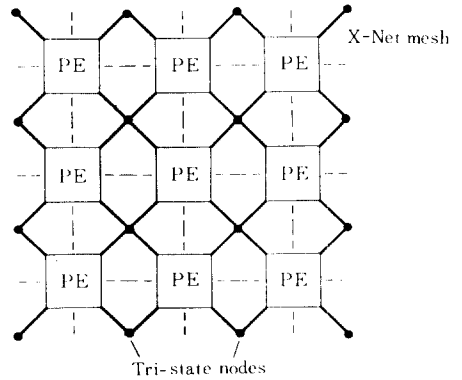


그림 3. X-net mesh 연결

IV. 공유 메모리 방식 병렬 컴퓨터

메모리 공유 MIMD 컴퓨터는 하나의 단일한 주소 영역을 갖는 공유 메모리를 가진 수 개 혹은 수 십개의 PE들이 공유 메모리를 통해 서로 협력하며 연산을 수행하는 구조의 컴퓨터이다. 그러므로, 이 방식의 특징은 메시지 전송 MIMD 구조와는 달리 모든 프로세서가 하나의 커다란 메모리 영역을 직접 액세스 할 수 있으므로, 사용자의 입장에서는 일반적인 단일 프로세서 컴퓨터와 유사한 프로그래밍 환경으로 인식 될 수 있는 장점이 있다. 반면에, PE들과 공유 메모리들 사이의 버스 병목 현상(bottleneck)을 없애 주기 위한 복잡한 스위칭 네트워크, 각각의 PE에 연결된 캐쉬 메모리들 사이의 상호 일관성을 유지하기 위한 하드웨어 그리고 메모리의 대역폭을 확장시키기 위한 복잡한 메모리 구조등 하드웨어의 개발에 소요되는 기술, 비용등이 메시지 전송 방식의 MIMD 컴퓨터 보다 훨씬 어렵고 많이 드는 단점이 있다.

메모리 공유 방식의 MIMD 구조는 크게 공유 버스 구조와 다단 네트워크 구조의 두 가지로 대별 될 수 있는데 전자의 대표적인 예로는 CMU-Cm, Elexsi, Sequent사의 Balance와 Symmetry 등을 들 수 있으며, 후자는 NYU Ultracomputer, Cedar, BBN Butterfly, IBM RP3, Alliant, Convex, FPS 등을 들 수 있다. 위의 전자의 구조는 주로 10개 이하의 대규모 PE들을 사용하고 있어 확장성에서 문제가 있으며, 현재는 주로 후자의 형태가 주로 연구되고 있다. 본 장에서는 이들 중 Alliant, BBN, Convex, FPS, Sequent 등의 컴퓨터의 구조 및 특성에 대해서 알아 보기로 한다.

1. Alliant Computer System

Alliant Computer System사는 병렬 처리, 3-D Graphic 그리고 industry standard를 기본으로 하는 FX/2800 series를 1990년 초기에 발표 하였다. 또한 Alliant와 Intel사는 i860 프로세서에 archicture extended (PAX) standard 개발에 참여하고 있다. FX/2800에서 프로세서와 공유 메모리는 interconnection network과 cache memory를 통하여 연결되어 있다. FX/2800은 4개의 i860을 갖는 연산모듈로 구성되며 7개 까지의 연산 모듈을 가질 수 있다. 이때 최대 연산 능력은 1.1 Gflops에 이른다. 각 연산 모듈의 4개의 i860중 2개는 super computational elements(SCE)로 사용되고 다른 2개는 super interactive processor(SIP)로 사용된다. SCE는 수치연산 전용이고 SIP는 오퍼레이팅 시스템과 사용자와의 연결기능을 포함한다. 한 개의 SIP를 갖고 있는 I/O 모듈은 20Mbyte/s 채널을 통하여

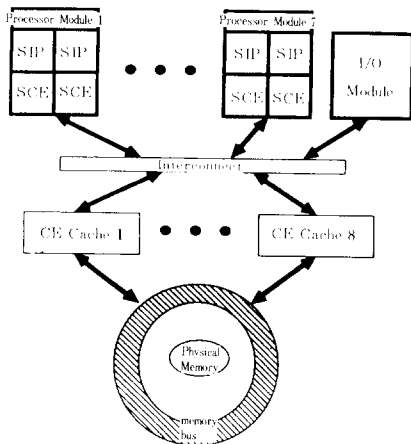


그림 4. Alliant FX/2800의 구조

외부와 연결되어 있다. 그림 4는 FX/2800의 구조를 나타낸다. FX/2800에서는 흔히 사용하는 방식인 각 프로세서마다 local cache를 갖도록 하는 분산 cache 방식 대신 공유가능한 interleaved cache를 사용하고 이를 interconnection network을 통하여 프로세서와 연결함으로써 여러 프로세서가 메모리 접근시 발생하는 충돌을 줄여서 성능을 개선하였다. 또한 cache 메모리를 통하여 프로세서간의 통신을 함으로써 통신 속도를 증가시켰다.

FX/2800에서는 오퍼레이팅 시스템으로서 UNIX의 일종인 concentrix를 사용하는데 병렬처리에 필요한 기능들이 포함되어 있다. 대개 여러 개의 프로세서는 오퍼레이팅 시스템에 의하여 시분할(time shared)로 여러 작업에 의해 사용되고 필요에 따라서 하나의 작업을 병렬 수행되도록 할 수 있다.

2. BBN Computer

BBN Advanced Computer사는 FFT에서 사용되는 butterfly와 유사한 구조를 갖는 butterfly라는 빠른 통신용 스위치 개발하여 컴퓨터 시스템에서 프로세서와 메모리 사이의 연결에 사용하였다. 1981년 68000 프로세서를 사용한 Butterfly 컴퓨터를 발표하였으며 68020을 사용한 Butterfly-Plus를 계속하여 발표하였다. 1989년에는 MC88000 계열의 RISC 프로세서를 사용한 TC2000 (time critical)를 발표하였다. TC2000 시스템의 핵심은 butterfly 스위치이다. Butterfly 스위치는 8×8 crossbar 스위치 모듈들로 구성되어 그림 5과 같이

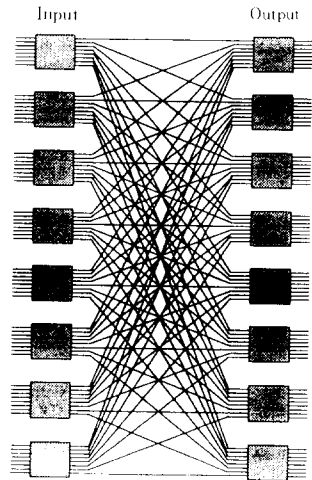


그림 5. 2단계 butterfly switch

서로 연결되어 있다. 그림 5와 같이 2단계로 연결한 경우에는 최대 64개 프로세서들을 연결할 수 있고 3단계로 연결한 경우에는 최대 512개의 프로세서들을 연결할 수 있다. 각 프로세서는 메모리의 위치에 관계없이 모든 메모리와 butterfly 스위치를 통하여 연결이 된다. 대부분의 병렬처리 컴퓨터는 프로그램을 개발하는 host 컴퓨터와 병렬처리 컴퓨터를 분리한다. 그러나 TC2000에서는 오퍼레이팅 시스템을 직접 각 프로세서에서 수행 할 수 있도록 하였다. TC2000에서는 nX 그리고 pSOS의 두 개의 오퍼레이팅 시스템이 있으며, nX는 UNIX를 이용한 다중 사용자를 위한 오퍼레이팅 시스템이고, pSOS는 실시간 응용에 적합하도록 설계되어 있다. TC2000에서는 각 프로세서에서 서로 다른 작업을 할 수 있으며, 이 시스템을 분할하여 독립된 cluster로 사용하여 여러 작업을 동시에 수행 할 수도 있다.

3. Convex Computer

Convex Computer사는 병렬처리 컴퓨터인 C201, C202에 이어 속도 면에서 개선된 C210, C240을 발표하였다. C240은 두 개, 세 개 또는 네 개의 프로세서를 직접 200Mbyte/s 시스템 버스를 통하여 직접 공유 메모리에 연결되어 있다. C240의 구조는 그림 6과 같다. 한 메모리 블록이 다른 메모리 블록에 의하여 교체될 때만 cache 메모리에 원하는 메모리를 복사하는 두 준위의 cache 시스템의 단점을 보완하기 위하여 명령어, 주소, 데이터 cache로 분리하여 cache 시스템을 구성하였다. Convex Computer사는 컴퓨터 시스템의 병렬성을 관찰하는 automatic self allocating processors(ASAP)라고 불리는 하드웨어 장치를 개발하였다. ASAP는 하드웨어 스케줄러를 사용하여 활용 가능한 프로세서에 작업을

할당하여 프로세서가 휴지 상태가 될 확률을 줄임으로써 전체 프로세서의 활용도를 극대화 할 수 있도록 하였다. ASAP는 프로세서의 명령어와 프로세스들을 프로세서들에게 모든 프로세스들이 종료될 때까지 공급한다. ASAP는 fork, schedule과 join의 3단계로 구성된다. 첫 번째는 모든 프로세서가 공유하는 통신 레지스터에 fork 요구를 기록한다. 그 다음에 scheduler는 통신 레지스터의 기록을 보고 휴지하고 있는 프로세서를 fork 요구한 프로세스에게 할당한다. 프로세스의 수행 결과는 통신 레지스터를 통하여 돌아온다. ASAP를 이용하여 fork, schedule된 프로세스는 전체 공유 메모리를 통하여 통신을 한다. Convex의 오퍼레이팅 시스템은 병렬 처리에 적합하도록 개선된 UNIX를 사용한다. 이 오퍼레이팅 시스템에서 병렬처리에 적합한 특징들은 하드웨어 장치, ASAP 장치, 세마포어 기술을 이용하여 구현되었다.

4. FPS Computing

FPS Computing사는 1988년에 Model 500을 발표하였다. Model 500은 스칼라, 벡터 프로세서를 갖는 모듈러 시스템이다. 이 모듈러 특성은 scalable interconnection architecture(SIA)를 이용하여 구현 하였으며, 이것이 Model 500의 핵심 기술이다. 이 모듈러 특성을 이용하여 사용자는 전체적인 하드웨어의 변경없이 시스템 확장을 용이하게 할 수 있다. SIA는 33MHz에서 64비트 데이터 패스로 구성되어 대역폭이 267Mbyte/s인 것과 이와 동일한 데이터 버스를 병렬로 4개를 이용하여 대역폭을 1Gbyte/s 이상으로 구성하는 방법이 있다. 1989년 벡터 프로세서를 개선한 Model 500EA를 발표하였다. 또한 1990년에는 SPRAC 칩을 사용하는 System 500을 개발하였으나 아직 상품화는 이루어지지 않고 있다. 최근에 발표된 스칼라와 벡터 유닛을 대치 할 수 있는 matrix co-processor를 이용하여 system 500 matrix co-processor를 구성하였다. 각 matrix co-processor는 4개에서 84개의 i860을 가질 수 있다. 그러므로 System 500 구조에서는 최고 13.4 Gflops의 연산 능력을 갖는다. 이 시스템은 7개의 버스 와 메모리 연결을 위한 8x8 crossbar 스위치가 있으며 그림 7과 같이 구성되어 있다.

시스템 메모리와 연결하기 위하여 matrix co-processor는 32 혹은 64Mbyte matrix 레지스터를 갖고 있다. 메모리 시스템은 동시에 연결될 수 있도록 8개의 독립된 포트 로 구성되어 있다. System 500에서는 SPARC 칩에서 작동하는 Sun OS를 오퍼레이팅 시스템으로 사용한다.

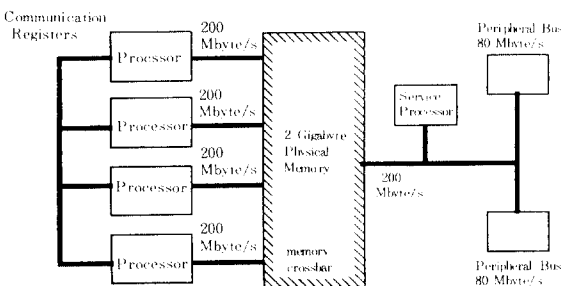


그림 6. Convex C204의 구조

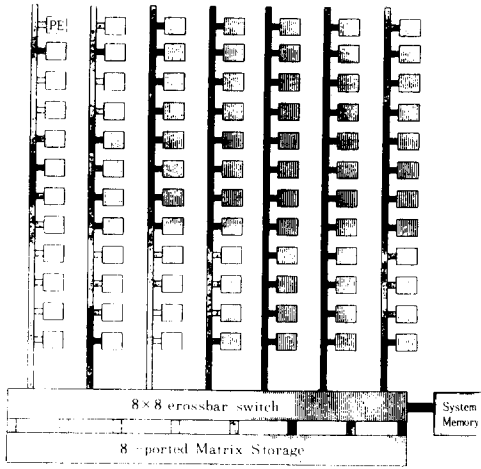


그림 7. FPS System 500 matrix co-processor

5. Sequent Computer Systems

Sequent사의 컴퓨터들은 프로세서들 간의 통신을 동 일 버스에 연결된 공유 메모리를 통해서 하는 전형적인 MIMD 방식의 컴퓨터이다. Sequent는 1984년에 UNIX 를 기반으로 2개에서 12개까지의 프로세서들을 자체의 시스템 버스를 이용하여 연결한 MIMD 컴퓨터로서 소개 되었다. Sequent사의 컴퓨터의 다중프로세서 구조는 그림 8과 같다.

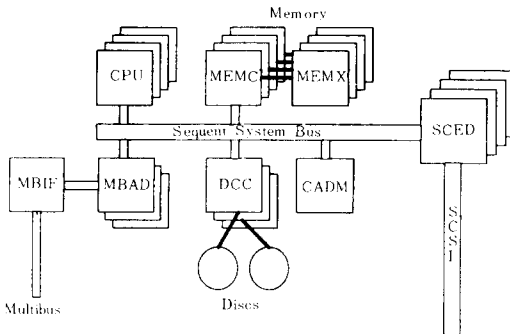


그림 8. Sequent의 다중프로세서 구조

Sequent사는 현재 NS32032를 사용한 Balance(B8 and B2)와 80386을 사용한 Symmetry(S27 and S81)라는 두 종류의 컴퓨터 시스템을 판매하고 있으나, 두 종류 모두 구조는 그림 8과 같이 구성되어 있다. Sequent사가 설계 한 system link controller(SLIC) 칩을 사용하여 프로세

서와 프로세서 그리고 다른 장치들과의 통신을 관장한다. 여러 프로세서와 기타 디바이스들이 동시에 시스템 버스를 사용하고자 할 때 생기는 충돌 문제는 controlled clock/arbitration(CACD) 보드가 제어한다. 시스템 버스를 사용하는 컴퓨터의 여러 자원들은 각기 우선 순위가 있다. 프로세서 보드는 낮은 우선 순위를 가지며, SCSI/ethernet/diagnostic(SCED) 보드, dual disc controller(DDC), multibus adaptor와 같은 주변 제어기 들은 높은 우선 순위를 갖는다. Sequent사는 DYNIX라는 UNIX에 병렬구조를 첨가하여 구성한 다중 처리 오 퍼레이팅 시스템을 제공한다.

V. 메시지 전송 방식 병렬 컴퓨터

메시지 전송 방식 MIMD 컴퓨터는 공유 메모리가 없 이 각각의 PE가 그 자신만의 로컬 메모리를 갖고 분할 된 일의 수행을 여러 형태의 통신 링크를 이용하여 서로 협력하여 수행해 나가는 방식의 구조를 갖는다. PE들 의 연결 형태는 하이퍼큐브, 메쉬, 스타, 피라미드, 토 러스 등의 다양한 형태로 이루어 지는데 그 연결의 확일 성을 이용하여 수 천에서 수 만개까지의 PE를 가진 거 대한 병렬 컴퓨터를 쉽게 만들 수 있는 장점을 가진다. 또한 각 PE의 구조는 일반적인 소형의 단일 프로세 서 시스템에 통신용 하드웨어만의 접속으로도 제작 이 가능하여 하드웨어적인 측면에서의 개발이 용이하 다. 반면에, 단일 응용 프로그램을 각 PE에 분할하 여 할당하는 작업의 어려움으로 인해 소프트웨어적 인 측면에서는 메모리 공유형 MIMD 방식보다도 훨 씬 불리한 단점을 안고 있다.

소프트웨어 측면에서의 단점에도 불구하고, 메시지 전송 방식 MIMD 컴퓨터는 현재 전 세계적으로 가장 널 리 연구되고 있는 병렬 처리 컴퓨터이며 현재 국내에서 도 널리 연구되고 있다. 본 장에서는 하이퍼큐브 컴퓨터 인 Intel사의 iPSC, NCUBE와 네트워크의 형태를 변환 가능한 transputer를 이용한 컴퓨터등에 대하여 알아보 고, 최근 Intel사가 발표한 바 있는 iPSC/DELTA 컴퓨 터에 대하여서도 알아 보겠다.

1. iPSC 하이퍼큐브 컴퓨터

미국의 Intel사는 가장 활발히 하이퍼큐브 컴퓨터를 발표하고 있는 회사이다. 1986년에 80286을 사용한 iPSC/1과 1988년에 80386을 사용한 iPSC/2를 발표한

이후 1990년에는 iPSC/860을 발표하였다. iPSC/1은 ethernet link로 만들어진 통신 채널을 통하여 통신을 하고 store-and-forward 라우팅 방식을 사용한다. 그리고 NX라는 작은 커널이 있어서 메시지 전송을 위한 기능을 제공한다. iPSC/2는 통신 채널로서 Caltech에서 개발된 웜홀(warmhole) 라우팅 방식을 구현한 2.8Mbyte/s의 전송속도를 갖는 direct-connet 라우팅 채널을 사용한다. 이 방식은 메시지의 헤더(header)는 패킷 스위칭 방식으로 전달하고 메시지의 나머지 부분은 두 노드 사이에 연결이 된 후 circuit switching 방식으로 전달하는 혼합스위칭 방식을 사용한다. Circuit 스위칭을 사용할 경우에는 메시지의 전송이 파이프라인 방식으로 수행되기 때문에 메시지 전송 시간이 store-and-forward 방식에 비해서 크게 단축이 된다. 통신 채널 중에서 하나는 I/O용 채널로 정하여 저장기능을 비롯한 여러 I/O기능을 가진 I/O노드와 연결할 수 있도록 하고 커널은 concurrent 파일 시스템(CFS)을 추가한 NX/2를 사용한다. iPSC / 860은 최신의 64-bit 프로세서인 i860을 사용하였고 나머지 특징은 iPSC/2와 비슷하다. iPSC/1, iPSC/2와 iPSC/860은 모두 최대 128 노드의 하이퍼큐브 컴퓨터를 제공하는데 iPSC/860은 최대 연산 속도가 7.6 Gflops에 이른다. iPSC를 위한 오퍼레이팅 시스템은 NX 오퍼레이팅 시스템 이외에도 다른 회사에서 MACH, Express, CrOS-III 등이 개발되어 제공되고 있다. 그림 9는 iPSC/860의 블럭 다이어그램이다.

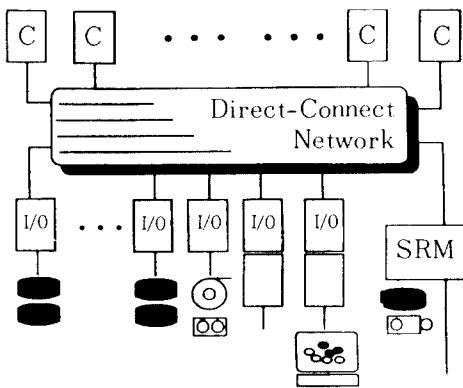


그림 9. Intel iPSC/860의 구조

2. NCUBE 하이퍼큐브 컴퓨터

NCUBE사는 1985년에 최초로 상용 하이퍼큐브 컴퓨터인 NCUBE/ten을 발표하였고 1989년에는 NCUBE/2

를 발표하였다. NCUBE 하이퍼큐브 컴퓨터는 자체 설계된 프로세서를 사용하는데 이 프로세서에는 32-bit 또는 64-bit의 프로세서와 부동소수점 연산장치와 통신용 채널이 포함되어 있다. 이러한 고집적 다기능 프로세서의 사용으로 인해서 하나의 프로세서 보드에 NCUBE/ten은 16개의 프로세서가, NCUBE/2는 32개의 프로세서가 있어서 커다란 하이퍼큐브 컴퓨터의 구성이 가능하다. NCUBE/2는 노드당 14개의 채널이 제공되어서 I/O용으로 사용하는 한 채널을 제외한 나머지 채널을 사용하여 최대 8192개 노드의 하이퍼큐브 컴퓨터를 구성할 수 있다. NCUBE/ten은 store-and-forward 라우팅 방식을 채택하고 있으며 NCUBE/2는 프로세서에 하드웨어 라우팅 기능을 내장하여 웜홀 라우팅을 할 수 있도록 하였다. 오퍼레이팅 시스템은 호스트용으로 UNIX의 일종인 Axis를 사용하는 데 하이퍼큐브의 서브 큐브는 UNIX device로서 간주할 수 있는 기능을 제공한다. 노드용 오퍼레이팅 시스템은 프로그램 로딩 기능과 메시지 전송 기능을 주로하는 작은 크기의 Vertex를 사용한다.

3. iPSC/DELTA

Intel사는 현재 DARPA의 Touchstone 프로젝트를 추진중인데 528개의 노드로 구성된 iPSC/DELTA를 발표했으며, 2000개 이상의 노드로 구성되는 iPSC/SIGMA와 iPSC/Teraflops를 진행중이다. iPSC/DELTA는 노드들이 메쉬 형태로 서로 연결된 메쉬 컴퓨터로서 570개의 프로세서를 사용하는데 이 중 528개는 i860으로 수치 연산용으로 사용하고 있으며 나머지는 80386으로서 I/O 처리용으로 사용하고 있다. 그림 10은 iPSC/DELTA 시스템의 구조를 나타낸 것으로서 프로세서들은 16×35의 메쉬 형태로 연결되어서 양쪽 끝열은 disk용 노드가 위치하고 가운데의 33열에 수치연산용 노드가 위치한다. 그리고 10개의 기타 I/O 기능을 하는 노드들이 연결되어 있다. iPSC/DELTA의 최대 연산 속도는 32 Gflops에 이르며 통신 채널은 하드웨어적으로 라우팅을 제어하고 circuit 스위칭을 한다. 노드의 갯수가 커짐에 따라서 이들을 서로 연결하는 복잡도가 급격히 증가하고 하이퍼큐브 형태로 연결할 수 있는 것에는 한계가 있어서 어느 정도 이상 크기의 하이퍼큐브 형태로 연결하는 것은 불가능하다. 이에 반해서 메쉬 형태로 연결하는 것은 이러한 한계가 거의 없으며 간단하다. 더욱이 하드웨어적으로 복잡한 면은 있지만 통신을 할 때에 circuit switching을 사용하면 하이퍼큐브와 비슷한 통신 성능을 얻을 수 있다.

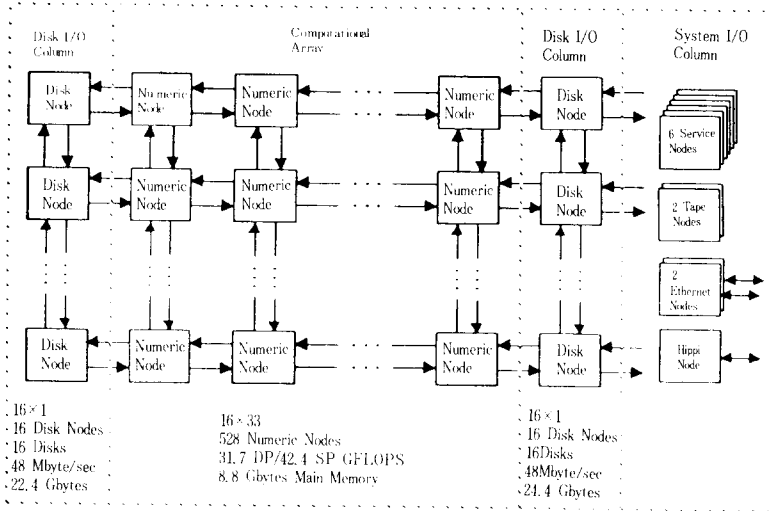


그림 10. iPSC/DELTA 시스템의 구조

4. Transputer를 사용하는 컴퓨터

Transputer는 Hoare의 CSP(communicating sequential processes)에 기초를 둔 OCCAM을 수행 모델로 하여 영국의 Inmos사에서 개발된 32-bit 프로세서이다. Transputer는 메시지 전송기능을 포함하여 최소의 회로로서 통신 기능을 수행할 수 있도록 하였고 이에 따라서 낮은 가격으로 비교적 높은 성능을 얻을 수 있게 해 준다. Transputer는 여러 종류의 프로세서가 있으며 그 중 32-bit 프로세서인 T414와 부동소수점 연산 장치를 포함한 T800이 가장 많이 이용된다. 그림 11은 T800의 내부 구조로서 4Kbytes의 on-chip RAM과 DMA 컨트롤러에 의해서 제어되는 20Mbps의 4개의 양방향 시리얼 링크를 갖고 있다. Inmos에서는 C104라는 32x32 crossbar switch 기능을 하는 스위칭 칩을 제공하는데 이 스위치 내에는 워홀 라우팅을 수행하여 효율적인 라우팅을 할 수 있도록 한다. 이것을 사용하면 여러 형태의 메시지 전송 방식의 병렬 컴퓨터를 용이하게 구현할 수 있다.

Transputer를 사용한 시스템은 PC, VMEbus 등에서 사용할 수 있는 부가 보드로서 많이 발표되었고 독립된 시스템으로서는 Parasytec의 Super Cluster 등이 있다. 그림 12는 64-노드의 Super Cluster의 구조를 나타낸다. Transputer는 노드의 프로세서로서 사용될 뿐만 아니라 i860 같은 고성능의 프로세서를 수치 연산용 프로세서로 사용할 경우 통신 기능을 전담하는 통신용 보조 프로세서로 사용되기도 한다.

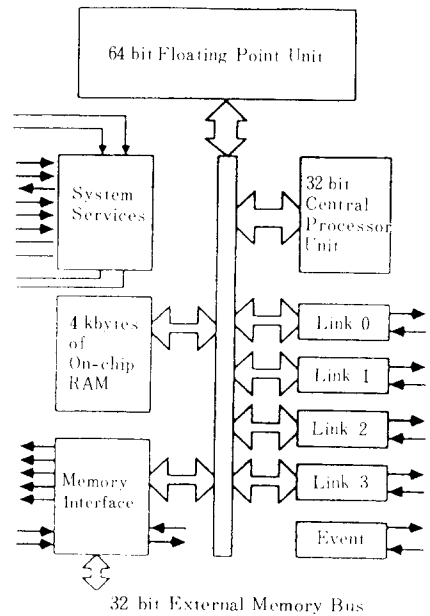


그림 11. Inmos T800 트랜스퓨터의 구조도

5. KAICUBE

KAIST에서는 국내에서 처음으로 1989년에 8 노드의 하이퍼큐브 컴퓨터인 KAICUBE-I을 발표한 데 이어 1990년에는 32 노드의 하이퍼큐브 컴퓨터인

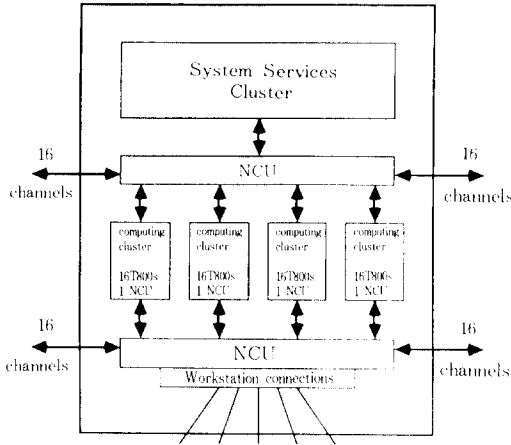


그림 12. Parasytec Super Cluster의 구조

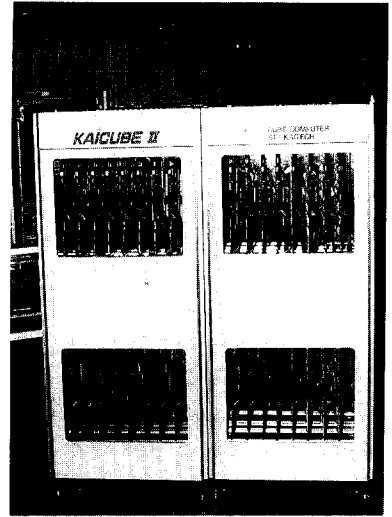


그림 13. KAICUBE-II의 외관

KAICUBE-II를 개발하였다. KAICUBE-II는 호스트 컴퓨터와 노드 컴퓨터로 구성되어 있다. KAICUBE-II의 노드 컴퓨터는 수치 연산용으로 MC68020 프로세서를 사용하고 통신을 전담하는 MC68000 프로세서를 별도로 사용하여 통신 능력을 보완하였다. 노드 컴퓨터의 오버레이팅 시스템으로는 일반적인 MIMD에 사용가능하도록 자체 설계, 개발한 커널을 사용하는 데 메시지 전송 기능을 비롯한 하이퍼큐브 컴퓨터에서 병렬 처리에 필요한 기능들을 제공한다. 통신은 store-and-forward 라우팅 방식을 사용한다. 호스트 컴퓨터로는 Motorola사의 68030 프로세서를 사용한 SYS-1132 시스템을 사용한다. 호스트와 노드 컴퓨터와의 통신을 위한 VME 버스에 통신용 인터페이스 보드를 제작하여 부가하였고 UNIX에 이를 사용하기 위한 device driver를 추가하였다. 호스트 컴퓨터에 하이퍼큐브 컴퓨터를 여러 사용자가 효율적으로 사용할 수 있도록 하기 위하여 다중 사용자 환경을 제공하였다. 현재는 최신 프로세서인 i860을 사용하여 KAICUBE-III 하이퍼큐브 컴퓨터를 개발중에 있다. KAICUBE-III는 최대 128 노드 하이퍼큐브를 구성할 수 있도록 설계되었다.

VI. 결 론

벡터 프로세서를 이용한 슈퍼 컴퓨터 개발의 성능 향상은 물리적인 한계로 인해 제한을 받는다. 이에 비해서 다수의 프로세서를 사용하는 병렬 컴퓨터가 VLSI 설

계, 제조 기술의 급속한 발전과 더불어 증가하고 있다. 병렬 컴퓨터는 프로세서의 수를 증가시킴으로써 성능 향상을 얻을 수 있는데 기술의 발전에 따라서 병렬 컴퓨터에 사용되는 프로세서의 수는 계속해서 증가할 것이다. 프로세서의 수가 증가할수록 가장 중요한 작업은 각 프로세서 사이의 통신 속도를 향상시켜 많은 프로세서간의 데이터 교환과 동기를 빠른 시간내에 이룰 수 있게 해야 한다.

병렬 컴퓨터는 일반적으로 널리 사용되기 위해서는 소프트웨어적인 면에서 많은 보완이 필요하다. SIMD 방식의 병렬 컴퓨터는 응용 범위가 제한되어서 일반적인 목적을 위해서 사용되기는 쉽지 않기 때문에 특정 응용 분야에서 최대한의 성능을 발휘할 수 있도록 발전시키는 것이 바람직하다. MIMD 방식의 병렬 컴퓨터는 SIMD 방식에 비하여 범용으로 사용할 수 있다. 그러나 여러 가지 소프트웨어의 미비로 사용하기가 쉽지 않다. 현재 표준화된 시스템 소프트웨어와 사용자를 위한 프로그래밍 환경이 없고 각 회사마다 각기 다른 구조의 소프트웨어 환경을 제공한다. 각 병렬 컴퓨터의 구조에 따라서 별도로 병렬 프로그램을 작성하는 일이 어렵다. 다양한 병렬 컴퓨터에서 동작할 수 있는 병렬 소프트웨어의 축적과 발전을 위해서는 하부의 병렬 컴퓨터 구조와 상관없이 병렬 프로그램을 개발하는 방법이 연구되어야 한다. 이것은 소프트웨어와 하드웨어와의 다른 구조

로 인한 부가 비용이 첨가되는 단점이 있다. 무엇보다도 중요한 일은 순차적 환경에 익숙한 사용자에게 하드웨어와 소프트웨어의 병렬성을 최대한 감추는 프로그래밍 환경이 제공될 수 있도록 하여야 한다. 즉, 병렬 프로그래밍에 있어서 많은 부분들을 시스템 소프트웨어나 유틸리티 프로그램이 처리할 수 있도록 연구되어야 한다. 기존에 개발된 많은 과학 기술용 순차적 프로그램을 병렬 프로그램으로 변환시켜줄 수 있는 변환 프로그램의 개발도 앞으로의 중요한 분야가 될 것이다.

현재까지의 병렬 컴퓨터의 주종은 공유 메모리 방식의 컴퓨터이다. 이 방식은 병렬 프로그램의 수행목적 이외의 범용으로서 사용하기에는 메시지 전송 방식보다는 이점이 있다. 그러나 사용가능한 프로세서의 수가 메시지 전송 방식에 비해 적기 때문에 더욱 증대되는 고속 계산의 요구를 충족시켜 주기 위해서는 메시지 전송 방식의 병렬 컴퓨터가 유망할 것으로 보인다. 병렬 컴퓨터의 이용가치를 높이기 위해서는 병렬 컴퓨터의 설계 기술의 향상으로 더 빠른 계산 속도를 제공하고, 표준화된 시스템 소프트웨어와 병렬 컴퓨터를 쉽게 사용할 수 있도록 하는 사용자 환경을 개발해야 하며, 효과적인 사용을 위해서 많은 사용자들이 병렬 프로그래밍에 익숙해지는 것이 필요하다.

參 考 文 獻

- [1] G. S. Almasi, and A. J. Gottlieb, *Highly Parallel Computing*, Benjamin Cummings, 1989.
- [2] W. C. Athas and C. L. Seitz, "Multicomputers : Message-passing concurrent computers," *IEEE Computer*, vol. 21, no. 8, pp. 9-24, Aug. 1988.
- [3] G. Fox et. al., *Solving Problems on Concurrent Processors*: vol. 1, Prentice-Hall, 1988.
- [4] P. Hayes, T. Mudge, Q. Stout, S. Colley and J. Palmer, "Architecture of a Hypercube Supercomputer," Proc. 1986 Int. Conf. on Parallel Processing, pp. 653-660, 1986.
- [5] W. D. Hills, *The Connection Machine*, MIT Press, Cambridge, MA, 1985.
- [6] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*, McGraw-Hill, 1988.
- [7] Intel Scientific Computers, iPSC/860 brochures.
- [8] NCUBE Corporation, NCUBE/2 brochures.
- [9] Paracom, Transputer products brochures.
- [10] G. F. Pfister, et. al., "The IBM Research Parallel Processor Prototype (RP3) : Introduction and Architecture," Proc. of 1985 Int. Conf. on Parallel Processing, pp. 764-771, Aug. 1985.
- [11] C. L. Seitz, "The cosmic cube," *Comm. ACM*, vol. 28, no. 1, pp. 22-33, 1985.
- [12] S. Thakkar, et. al., "The balance multiprocessor system," *IEEE Micro*, pp. 57-69, Feb. 1988.
- [13] A. Trew and G. Wilson, *Past, Present, Parallel: A Survey of Available Parallel Computing Systems*, Springer-Verlag, 1991. 

筆者紹介



金 明 桓

1932年 2月 8日生
 1958年 University of Alabama,
 전기공학(학사)
 1959年 Yale University,
 전기공학(석사)
 1962年 Yale University,
 전기공학(박사)

1961年~1962年 New Haven College, 전기공학과
 강사
 1962年~1983年 Cornell University 전기공학과 교수
 1967年~1968年 Johnson NASA 연구소 초빙교수
 1968年~1969年 Jet Propulsion Laboratory
 선임연구원
 1969年~1970年 California Institute of Technology
 초빙 부교수
 1972年~1977年 US Army Missile Command
 초빙연구원
 1989年~1990年 생산기술연구원 부원장
 1983年~현재 한국과학기술원 전기 및 전자공학과
 교수
 주관심분야 : 실시간 분산처리, 병렬처리, VLSI 구조
 및 설계, 데이터베이스



尹 相 均

1961年 9月 25日生
 1984年 서울대학교 공과대학
 전자공학과(학사)
 1986年 한국과학기술원 전기 및
 전자공학과(석사)
 1990年~현재 한국과학기술원
 전기 및 전자공학과
 (박사과정)

1984年~1990年 현대전자산업(주) 컴퓨터사업본부
 주관심분야 : 컴퓨터구조, 병렬처리, VLSI 구조 및
 설계