

병렬처리를 이용한 화력발전소의 실시간 시뮬레이션

Real Time Simulation using Multiple DSPs for Fossil Power Plants

°박희준*, 김병국**

* 한국과학기술원 전기및전자공학과(Tel : +82-42-869-5435; E-mail:hjpark@cclab.kaist.ac.kr)

** 한국과학기술원 전기및전자공학과(Tel : +82-42-869-3435; E-mail:bkkim@cclab.kaist.ac.kr)

Abstracts A fossil power plant can be modeled by a lot of algebraic equations and differential equations. When we simulate a large, complicated fossil power plant by a computer such as workstation or PC, it takes much time until overall equations are completely calculated. Therefore, new processing systems which have high computing speed is ultimately needed to develop real-time simulators. Vital points of real-time simulators are accuracy, computing speed, and deadline observing. In this paper, we present a enhanced strategy in which we can provide powerful computing power by parallel processing of DSP processors with communication links. We designed general purpose DSP modules, and a VME interface module. Because the DSP module is designed for general purpose, we can easily expand the parallel system by just connecting new DSP modules to the system. Additionally we propose methods about downloading programs, initial data to each DSP module via VME bus, DPRAM and processing sequences about computing and updating values between DSP modules and CPU30 board when the simulator is working.

Keywords parallel processing, multiple DSP, real time simulation, power plant

1. 서론

화력발전소를 수학적으로 모델링할 경우 전체 시스템은 보일러, 스팀터빈, 응축기, 급수기, 발전기, 가스 Merge and Split 등 여러 부분의 작은 시스템으로 나뉘어 질 수 있고 각각의 시스템들은 미분방정식으로 나타내어질 수 있다[3]. 발전소 규모에 따라 많은 미분 방정식이 포함된 수식을 컴퓨터를 이용하여 계산하려고 할 때, 계산량이 많으므로 결과가 나오기까지 많은 시간이 필요하게 된다. 단순히 제어알고리즘의 검증과 수행결과가 목적이거나 계산시간이 중요하지 않지만 실제 발전소 운전원의 훈련을 위한 시뮬레이터를 구현하고자 할 때는 발전소 운전원이 실제 화력발전소와의 차이점을 거의 못 느끼도록 유사한 환경을 제공해야 한다[2]. 그러기 위해서는 정해진 샘플링 주기마다 주어진 계산량을 처리해 내는 실시간 시뮬레이터가 필요하게 된다. 실시간 시뮬레이터의 요건으로는 정확성, 빠른 계산능력, 완료시간(deadline)의 준수 등을 들 수 있다. 실시간 화력발전소 시뮬레이터 구현에 있어서 가장 큰 문제가 되는 많은 계산량을 병렬처리를 통해 해결하려는 시도는 Rafin[4]과 Hatcher[1] 등에 의해 이루어지고 있다. 이 논문에서는 Multiple DSP를 이용하여 실시간 화력발전소 시뮬레이터를 개발하고 병렬처리에 따른 여러 가지 문제점에 대한 해결방법을 제시한다. 시뮬레이터로 사용되기 위해 제작되는 하드웨어는 빠른 DSP 프로세서와 메모리를 가지고 있는 여러개의 DSP 모듈로 구성되고 각각의 DSP 모듈이 서로 통신하며 병렬처리를 수행하는 구조이다. 확장성을 고려하여 분산메모리구조로 설계하였으며 시뮬레이터와 제어기의 통신은 VME bus를 이용하게 된다. 이러한 구조의 시뮬레이터 개발시 각각의 DSP 모듈에 Program과 초기값을 download하는 문제, 각 DSP 모듈간의 적절한 작업량 분담, 시뮬레이션 수행시 통신을 통한 상태값 갱신에 대한 알고리즘 개발이 필요하다.

이 논문에서는 병렬처리를 이용한 화력발전소 시뮬레이터 제작시 생기는 이러한 문제에 대한 알고리즘을 제안한다.

2. 시뮬레이터의 구조

2.1 DSP 모듈의 제작

이 연구에서 사용될 DSP 모듈은 TMS320C40을 이용하였으며 다른 분야에서도 쉽게 확장, 응용할 수 있도록 확장성과 소형화를 고려하여 설계되었다. TMS320C40은 병렬처리를 위한 목적으로 설계된 프로세서로서 이를 이용한 병렬처리 시스템을 쉽게 구현할 수 있도록 많은 기능을 제공한다[5][6].

표 1 DSP 모듈의 구성
TABLE 1 DSP 모듈 overview

CPU	TMS320C40 50MHz, 50MFLOPS, floating point
RAM	SRAM 20ns, 1M bytes 0 wait access
ROM	EPROM 512K bytes
Connector	global bus, local bus
Communication port	6 ports 20M bytes per second

TMS320C40은 global과 local로 나뉘어진 각각의 버스를 가지고 있다. 이 버스는 31 bit address, 32 bit data, clock, timer, 4개의 interrupt, CPU의 bus access 상태를 나타내는 state, 그밖에 control선으로 구성되어 있다. 다른 I/O 모듈, 인터페이스 모듈, 확장메모리 모듈 등과의 확장성을 위하여 위의 모든 버스가 인이 두 개의 커넥터에 연결되어 있다. 이 연구에서는 이 버스 커넥터가 VME 인터페이스 보드와의 통신용으로 사용된다. 또한 범용 모듈로 제작되므로 소형화에 노력하였다. 모듈기판의 크기는 160mm x 100mm로서 PC의 ISA 보드 또는 VME 보드에 두 개의 모듈이 1층으로 고정될 수 있다. 이렇게 한 보드에 시뮬레이터의 모든 시스템을 고정시킴으로서 시스템을 소형화할 수 있고 안정성을 높일 수 있다. 표1은 DSP 모듈의 성능 및 특징을 나타낸다.

DSP 모듈간의 통신은 각각의 communication port를 커넥터로 연결함으로써 간단히 이루어진다. 이 communication port는 1개 모듈당 6개씩 제공되므로 병렬처리 시스템 구성시 선형, 평면, 트리, 링, 하이퍼큐브 등 다양한 구조를 실현시킬 수 있고 1초당 20M bytes의 전송속도이므로 데이터 및 프로그램을 고속으로 전송시킬 수 있다[6].

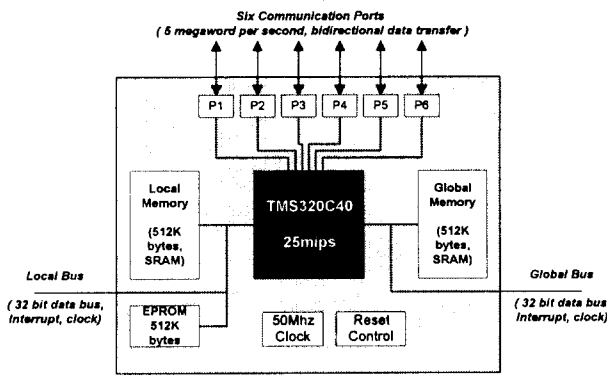


그림 1 DSP 모듈의 구조

Fig 1 Block Diagram of the DSP Module

2.2 VME 인터페이스 모듈

이 연구에서 제작되는 시스템은 제어기로서 CPU30 보드가 사용되고 시뮬레이터로 4개의 DSP 모듈이 사용되므로 VME 버스를 통한 통신을 하기 위하여 VME 인터페이스 모듈이 사용된다. 이 모듈은 1K words의 DPRAM과 DPRAM 접근을 증가해 주기 위한 Access Control Logic, 그리고 DSP 모듈의 디스플레이 수단으로서 8 bit LED로 구성된다. Access Control Logic은 양쪽의 버스가 동시에 같은 어드레스에 접근하지 못하도록 wait 신호를 발생시켜 주는 역할을 하여 안정적으로 DPRAM에 읽고 쓸 수 있게 해 준다.

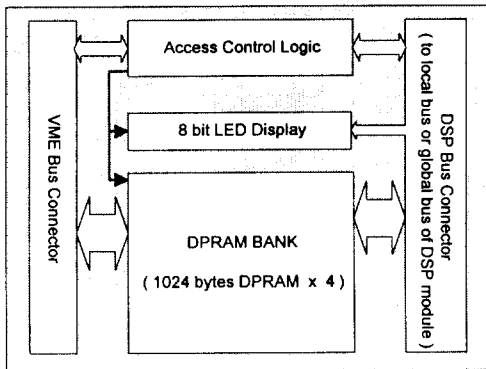


그림 2 VME 인터페이스 모듈의 구조

Fig 2 Block diagram of the VME Interface Module

2.3 전체 시뮬레이터 시스템의 구성

전체 시뮬레이터 시스템은 workstation을 Host Computer로 하고 CPU30 보드가 발전소 제어기로서, 그리고 4개의 DSP 모듈과 VME 인터페이스 모듈이 발전소 시뮬레이터의 역할을 하게 된다.

먼저 workstation에는 CPU30 보드와 4개의 DSP 모듈에서 수행될 프로그램을 개발하는 환경이 되며 시스템이 동작되기전에

다운로드할 프로그램이 저장되어 있는 곳이다. 또한 시뮬레이터와 제어기의 상태를 그래프로 디스플레이 하는 역할을 할 수 있다.

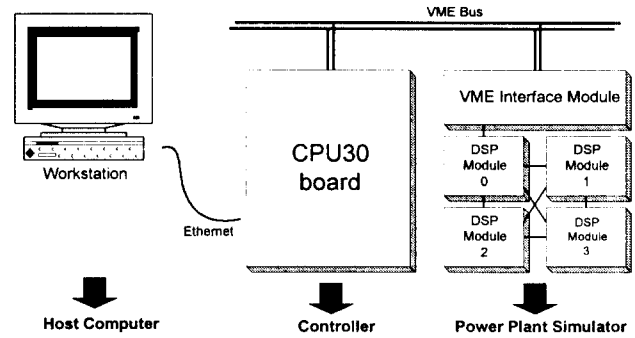


그림 3 전체 시스템 구조도

Fig 3 Overall System Architecture

CPU30 보드는 발전소 제어기 역할을 하며 workstation과는 ethernet을 이용하여 통신한다. socket을 사용하여 workstation과 통신하므로 어떠한 UNIX 시스템의 파일에도 접근할 수 있어서 UNIX 시스템을 '가상 파일 시스템으로 사용할 수 있다. 운영체제로는 vxWorks를 사용한다. vxWorks는 real-time OS로서 Unix와 호환되는 Network, Multitasking, Intertask Communication, Performance Evaluation Utility, Watchdog timer 등등 real-time 시스템을 개발하기에 좋은 환경을 제공하고 있다[7]. vxWorks의 타임 인터럽트 기능을 이용하여 sampling time을 정확히 지키면서 시뮬레이션을 지속할 수 있다. 시뮬레이터로서는 4개의 DSP 모듈이 고속 전송선에 의해 fully connected network을 구성하고 있다. CPU30과 DSP 모듈간의 데이터 통신 즉 각각의 DSP 모듈을 위한 프로그램과 발전소 초기값의 다운로드, 시뮬레이션 시 제어기와 입출력 값의 통신이 DPRAM을 통해 이루어진다. 그러므로 이러한 다양한 종류의 통신을 가능하게 하기 위하여 DPRAM을 이용한 통신 프로토콜의 개발이 필요하다.

3. DPRAM을 이용한 통신 프로토콜

CPU30과 DSP 모듈 #0는 DPRAM을 매개로 프로그램과 초기값을 다운로드한다. 시뮬레이션 수행시에는 시뮬레이터와 제어기의 입출력값을 교환하고 특정 어드레스를 flag로 사용함으로써 CPU30과 DSP 모듈이 동기화를 이루면서 작업을 수행하게 된다. 특히 실시간 시뮬레이션과 다중 프로세서간의 병렬처리시에는 동기화가 필수적이다. 이와 같이 DPRAM을 이용하여 다양한 작업들이 이루어지기 때문에 DPRAM을 체계적으로 사용하는 프로토콜이 필요하다.

프로토콜은 크게 두 가지로 나뉘어진다. 프로그램과 초기값을 다운로드 할 때의 경우와 시뮬레이션을 수행할 때이다.

먼저 다운로드 할 때의 경우에 대하여 살펴보면 CPU30이 다운로드할 데이터의 부분을 여러 부분으로 나누어서 헤더부분과 데이터부분으로 구성되는 패킷을 DPRAM에 기록하고 flag를 new로 set한다. 헤더에는 현재 패킷이 가지고 있는 데이터의 내용과 패킷번호, 패킷이 가지고 있는 유효한 데이터의 개수 등이 기록된다. DSP 모듈 #0는 flag를 polling하고 있다가 new가 읽혀지면 현재 패킷을 읽어서 헤더를 분석한 후 다른 DSP 모듈로 전송하거나 자신의 메모리로 다운로드 한다. 다운로드가 완료되면 flag를 clear하여 패킷을 읽었음을 CPU30에 알린다.

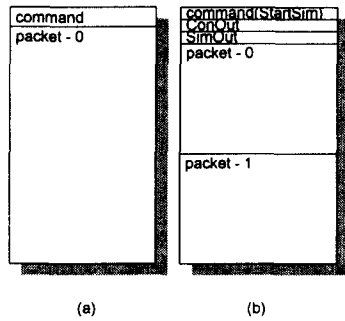


그림 4 DPRAM의 Memory Map (a) 다운로드 (b) 시뮬레이션
Fig 4 DPRAM Memory Map (a) Download (b) Simulation

제어기와 시뮬레이터가 실제 시뮬레이션 작업을 수행 중일 때는 쌍방향의 데이터전송이 이루어져야 하므로 DPRAM을 크게 두 개의 세그먼트와 세 개의 flag으로 나눈다. 세그먼트-0은 CPU30에서 DSP 모듈 #0로 전송되는 부분이고 세그먼트-1은 그 반대의 전송이 이루어지는 부분이다. 제어기와 시뮬레이터 사이에 교환되는 데이터 량은 상대적으로 작으므로 DPRAM을 두 영역으로 나누어도 충분히 한 패킷으로 전송 가능하다. 동기화에 사용되는 세 개의 flag은 StartSim, ConOut, SimOut으로서 각각의 flag이 set되어 있으면 StartSim은 DSP에 할당된 작업의 계산시작, ConOut은 제어기의 출력이 유효함 그리고 SimOut은 시뮬레이터의 출력이 유효함을 나타낸다.

4. 시뮬레이터와 제어기의 기동 절차

4.1 시뮬레이터 부팅

시뮬레이터의 4개의 DSP는 각각 자신의 EPROM에서 부팅프로그램을 읽어서 SRAM으로 다운로드한다. EPROM은 속도가 느려서 zero wait memory access가 불가능하기 때문에 속도가 빠른 SRAM으로 다운로드하는 것이다. DSP 모듈 #1, #2, #3의 부팅프로그램은 부팅이 성공적으로 끝났음을 나타내는 신호를 전송선을 통하여 DSP 모듈 #0에 알리고 대기상태로 들어간다. DSP 모듈 #0은 다른 DSP 모듈이 전송하는 부팅완료 신호를 DPRAM의 특정 어드레스에 기록한다.

4.2 CPU30 부팅

CPU30은 ethernet을 통해 workstation으로부터 OS인 vxWorks와 제어기 프로그램을 다운로드한다. 제어기 프로그램은 DPRAM을 읽어서 4개의 DSP 모듈이 모두 부팅완료되었는지 확인하고 만약 이상이 있으면 error처리 한다.

4.3 시뮬레이션 프로그램 다운로드

제어기 프로그램은 vxWorks의 가상 파일 시스템을 이용하여 workstation으로부터 DSP 모듈 #0, #1, #2, #3에 다운로드될 시뮬레이션 프로그램을 읽어서 다운로드를 시작한다. 이 작업은 CPU30이 DPRAM에 시뮬레이션 프로그램을 쓰면 DSP 모듈 #0이 이것을 읽어서 다운로드될 DSP 모듈로 전송해 주는 작업으로서 DPRAM의 용량에 비해 상대적으로 큰 프로그램을 다운로드 할 때 여러번에 나누어서 전송해야 되며 어떠한 프로그램이 4개의 DSP 모듈 중 어느 것으로 전송되어야 하는 지 알수 있어야 하기 때문에 앞에서 제시한 프로토콜이 사용된다.

다운로드가 완료되면 DSP 모듈 #1, #2, #3은 DSP 모듈 #0로 다운로드 완료 신호를 전송하고 이 신호는 DPRAM에 기록되어

CPU30이 다운로드가 성공적임을 확인한다.

4.4 발전소의 초기상태 다운로드

다음으로 발전소의 파라미터와 초기상태등의 변수들을 다운로드 하는 단계이다. 이 과정은 시뮬레이션 프로그램 다운로드와 유사한 과정으로 이루어진다.

4.5 시뮬레이터 기동

프로그램의 다운로드와 발전소의 파라미터, 초기상태의 다운로드가 완료되었다면 실제 제어기와 시뮬레이터가 각각의 입출력 데이터를 주고받으며 발전소 시뮬레이션을 하게 된다. CPU30, DSP 모듈 #0, DSP 모듈 #1, #2, #3의 작동 순서도가 그림5에 도시 되어 있다. 각 프로세서의 동작 순서는 크게 할당된 작업을 계산하는 부분과 다른 프로세서와 통신을 통하여 결과값을 갱신하는 부분으로 나뉘어 진다.

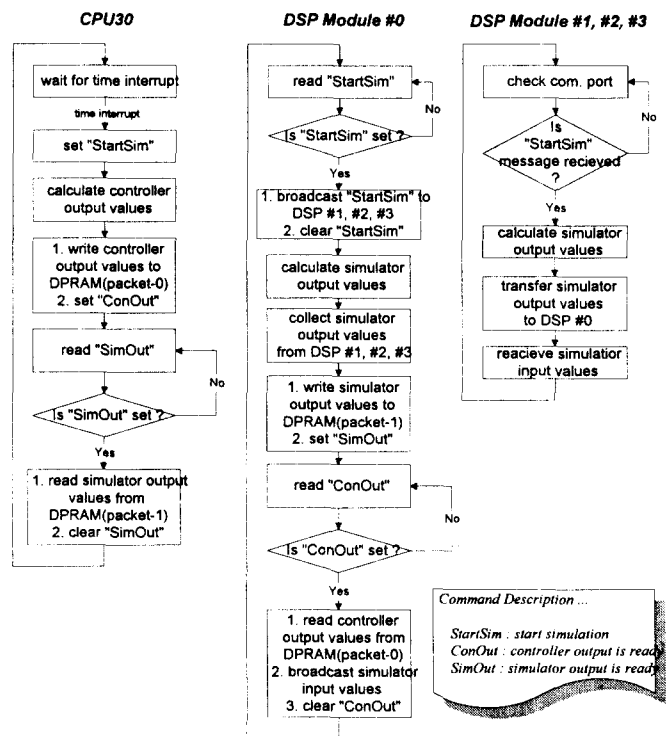


그림 5 CPU30, DSP #0, #1, #2, #3의 동작 순서
Fig 5 Operating Sequences of CPU30, DSP#0, #1, #2, #3

CPU30의 타임인터럽트에 의해서 CPU30은 StartSim을 set시키고 이 신호에 의해서 CPU30, DSP 모듈들이 할당된 작업에 대한 계산을 시작하게 된다.

계산이 완료되면 CPU30은 결과를 DPRAM에 기록한 뒤 ConOut을 set시키고 DSP모듈#0은 결과를 다른 모듈로부터 모아서 DPRAM에 기록하고 SimOut을 set시킨다. 이 두 flag의 상태에 따라서 set되어 있으면 계산결과를 읽어가고 clear시킨다. DSP 모듈들은 각각의 작업결과를 서로 갱신하고 난 뒤 StartSim을 기다리는 상태로 돌아간다. 이와 같은 방법으로 각각의 프로세서들은 동기화를 이루면서 작업계산 및 통신을 하게 된다.

5. 발전소 모델링과 작업할당

5.1 발전소 모델링

화력발전소는 여러 가지 종류가 있지만 크게 보일러, 스팀터빈, 응축기, 급수기, 발전기, 가스 Merge and Split 등의 하부 시스템으로 나뉘어 질 수 있고 각각의 시스템은 많은 대수적 수식과 상미분 방정식, 그리고 편미분 방정식으로 모델링된다. 많은 논문에서 발전소의 각 부분을 다양한 정확도를 가지는 모델링 방법을 제시하고 있다. [3]에서는 비선형상미분방정식을 바탕으로 모델을 기술하였고, [8]에서는 비선형상미분방정식 외에 편미분방정식도 도입하여 모델을 기술하였다.

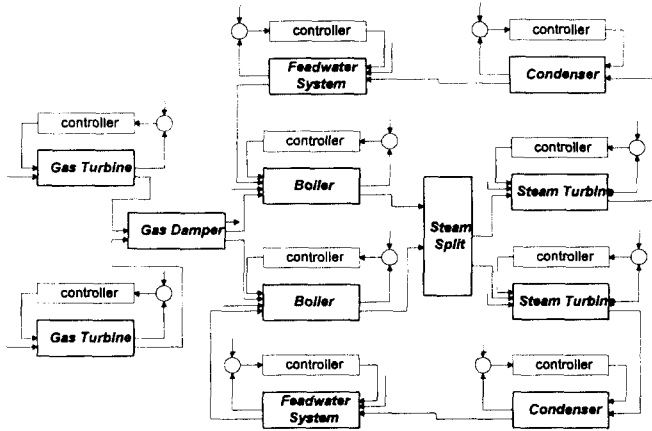


그림 6 CC/CHP configuration의 block diagram
Fig 6 CC/CHP configuration - block diagram

그림6는 [3]에서 발췌된 것으로서 화력발전소의 한 예이다. 이 연구에서는 이 모델을 시뮬레이션 대상으로 선정하였다. 이 발전소는 외부 사용자를 위해 전기와 스팀을 모두 생산한다. 가스터빈과 스팀터빈으로부터 발전하며 스팀은 외부사용자를 위해 고압과 저압의 형태로 나누어 공급된다. [3]에서는 이 모델에 대한 모델링된 수식도 수록되었다. 예를 들면 보일러는 42개의 대수적 수식과 15개의 상미분방정식으로 모델링되고 응축기는 13개의 대수적 수식과 4개의 상미분방정식, 5개의 편미분방정식으로 모델링된다. 상미분방정식과 편미분방정식은 수치해석적인 방법을 이용하여 C 언어로 기술되며 이것을 컴파일하여 각 DSP에 다운로드될 수 있다. 제어기는 P, PI, PD, PID의 조합으로 구성되므로 발전소 모델에 비하여 상대적으로 계산량이 적다고 할 수 있다.

5.2 4개의 DSP 모듈에 대한 작업할당

작업할당은 계산시간과 통신시간을 최소화하는 방향으로 진행되어야 한다. 모델링된 발전소의 각 하부 시스템들을 수행하는 데 걸리는 시간의 측정에는 vxWorks에서 제공되는 timexLib를 이용하여 간단히 측정할 수 있다. DSP 모듈에 수행시간을 측정하려는 함수를 다운로드시킨후 CPU30에서 실행시작을 알리는 메시지를 DPRAM에 씌어서 DSP는 함수 수행을 시작한다. 측정의 정확도를 높이기 위해서 정확한 해상도의 수행시간을 얻을 수 있을 만큼 함수를 반복수행 하는 방법을 이용할 수도 있다. 수행이 끝난 뒤 DPRAM에 수행완료를 알리는 메시지를 씌어서 수행시간 측정이 완료된다.

각 DSP 모듈간의 통신에 걸리는 시간은 비교적 간단히 계산할 수 있다. 즉 DSP 통신선의 속도와 각 DSP 모듈간에 전송해야될 변수들의 개수를 알면 한 개의 부동소수점 변수는 1 word 이므로 통신량과 앞에서 제시한 시뮬레이터 동작순서를 고려하여 계산할 수 있다.

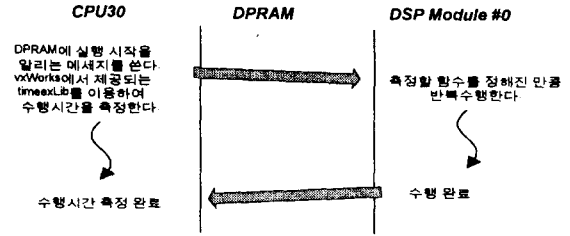


그림 7 DSP Module의 수행시간 측정
Fig 7 Measuring the Execution Time of DSP Module

실행시간과 통신시간의 합을 비용함수로 하여 이를 최소화 하는 작업할당 문제는 NP-Complete문제로 알려져있고 이를 해결하기 위한 많은 경험적 방법들이 연구되어 왔다. 하지만 이 연구에 사용되는 화력발전소는 몇 개의 많지 않은 하부시스템들로 이루어져 있고 각각의 하부시스템을 하나의 작업으로 본다면 단지 경험적인 고찰로서 4개의 DSP 모듈에 작업을 적절히 할당할 수 있다.

6. 결론

확장성과 소형화에 주력하여 설계된 DSP 모듈과 VME 인터페이스 모듈을 제작하여 4개의 DSP 모듈로 이루어진 병렬처리 시스템을 구현하였다. 그리고 이 시스템을 이용하여 많은 계산시간이 걸리는 화력발전소 시뮬레이션을 병렬처리를 통하여 실시간으로 수행하는 데 발생하는 동기화, 데이터통신, 작업할당 등에 대한 해결 방법을 제시하였다.

병렬처리 시스템은 사용되는 알고리즘과 작업의 할당방법, 그리고 통신방법에 따라 speedup에 차이가 생기므로 더 효율적인 방법에 대한 연구가 수행될 수 있고 다양한 병렬처리 방법에 따라 한 개의 DSP 모듈에서 수행할 때에 비하여 어느 정도의 speedup을 얻을 수 있는 지 정량적인 측정에 대한 부가적인 연구가 필요하다.

참고문헌

- [1] W. L. Hatcher, F. M. Brasch, Jr and J. E. Van Ness, "A Feasibility Study for the Solution of Transient Stability Problems by Multiprocessor Structures", IEEE trans. on PAS, Vol. PAS-96, no.6, pp1789-1797, 1977
- [2] S. Kyuwa and T. Yoshida, "Operator Training Simulator with Real-Time Transient Stability Analysis", IEEE trans. on Power Systems, Vol. 9, No. 2, pp721-729, 1994
- [3] A. W. Orders, "Modeling and Simulation of Power Generation Plants", Springer-Verlag, 1994
- [4] M. Rafian, M. J. H. Sterling and M. R. Irving, "Parallel processor algorithm for power system simulation", IEE Proc. Vol. 135, No.4, pp285-290, 1988
- [5] "TMS320c4x Parallel Processing Development System Technical Reference" Texas Instruments, 1993
- [6] "Parallel Processing with the TMS320c4x", Texas Instruments, 1994
- [7] "vxWorks Programmer's Guide", Wind River Systems, 1986
- [8] 김형래, "보일러 시스템의 열수력학적 동특성 시뮬레이션", 한국과학기술원 기계공학과, 1995