데이타 흐름 시스템을 이용한 호처리 프로세서의 구조

°임인택, 이성규, 한영철
삼성반도체통신 (주) 통신연구소

A New Architecture of Call Processor Based On Data flow System

IN TAEK LIM, SUNG GYU LEE, YOUNG CHUL HAN
Telecommunications Lab., Samsung Semiconductor & Telecommu. Co., Ltd.

Abstract

Conventional major electronic switching systems based on stored program control employ a Von Neumann styled control processor. It has strict limitations such that it essentially lacks concurrency in executing instructions, which have brought the software bottleneck problem, and the capabilities of call processing are restricted by expanding system's capacity.

In this paper, a new architecture of call control processor based on the data flow system is proposed, aiming at fundamental resolution for these limitations. The processor has a number of advantages in such as expansibility of system's capacity, parallel processing of calls, and so on.

1. 서 론

축적 프로그램제어(Strored Program Controlled) 교환방식을 이용하는 전자식 교환기 시스템의 상용화가 이미 실현된 이래에 교환기의 호처리(Call Processing)제어방식이Von Neumann 형태를 벗어나지 못한 상태에 놓여 있는 것이 현실적인 문제이다. 장래의 ISDN(Integrated Service Data Network)구축을 위한 회선수 증가에 따르는 통화량 혼잡 현상을 제거하고, 아울러 고도의 처리능력과 용이한 소프트웨어 개발을 위해서는 종래의 교환기 시스템에 사용되는 호처리 제어방식으로는 호의 병렬처리가 불가능하다. 또한 이는Von Neumann형태의 계산이론에 기초를 한 처리소자를 사용한다는 점에서 구현방법이 중앙 집중식 제어(1)이든 분산식 제어(2)이든 간에Von Neumann 형태의 기계가 갖는 제반 문제점을 가질 수 있다. 따라서 데이타 흐름 모델을 갖는 처리소자가 차세대 교환기 시스템의 호처리 제어부에 이용될 경우 데이타 흐름 모델 고유의 병렬처리 능력으로 인하여 교환기 내의Von Neumann 형태의 제한점이 극복되어 전반적인 성능향상을 꾀할 수 있다. (3,4)

뿐만아니라, 데이타 흐름 기에서는 처리소자의 수에 비례해서 처리 능력이 증가한다는 점을 고려한다면 가입자 수효의 증가에 따르는 확장이 용이하게 된다. 이러한 데이타 흐름 모델의 특성과 Von Neuman형태의 교환기 시스템의 제한점을 고려해서 본 논문에서는 호처리를 비동기식 병렬처리하기 위해서 데이타 흐름 모델을 이용한 새로운 교환기 시스템의 호처리 프로세서의 구조를 제안하고자 한다.

2. 데이타 흐름 시스템

데이타 흐름 모델은 연산에 필요한 입력 데이타가 존재함으로써 그 연산의 수행이 되는 계산이론으로써 Actor (또는 계산단위)와 Actor들 간의 데이타 종속성으로 이루어진다 (5, 6, 9). Actor는 기계 명령어의 집합으로된 계산단위를 나타내고, Actor의 출력이 다른 Actor의 입력이면 두 Actor는 데이타 종속성을 갖는다고 정의되며 이러한 데이타 종속성과Actor를 표현하는 방법으로는 데이타 흐름 그래프 (7)를 사용한다. 데이타 흐름 그래프는 노드(Node)와 아크(Arc)로 구성되는데 노드는 수행할 명령을 나타내고, 아크는 노드 사이의 데이타 전송관계를 나타낸다.

데이타 흐름 그래프의 수행규칙 (Firing Rule)은 그림2-1과 같이 노드의 모든 입력아크에 입력값이 존재하고 출력아크에는 어떤 값도 존재하지 않는 경우에 그 노드는 수행이 가능하다.

데이타 흐름 모델에 대한 여형론의 특성으로는 비동기적 동시성의 구현이 가능하다는 것인데, 이는 한 Actor의 필요한 모든 입력값이 존재하면 다른 Actor의 상태 또는 수행순서에 관계없이 수행이 가능하다는 것으로써, 한번 수행이 되면 결과 값은 그 값을 필요로

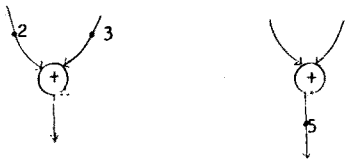


그림 2-1.수행규칙 (A) 수행전, (B) 수행후

하는 모든Actor 들로 분산되므로 참조투명성을 갖게 되어서 Side-effect 가 전혀 일어나지 않는다. 또한 공유 메모리의 개념이 없고, 프로그램 계수기(PC)가 필요없으므로 명령어 수행의 제약조건으로는 각 Actor 들간의 데이타 종속성만으로 정의가 된다.

3. 데이타 흐름 교환기 구조

교환기 통신망의 기능을 고도화하고, 경제적이고 신뢰성있는 시스템을 구현하면서, 아울러 효율적이고 경제적인 새로운 서비스를 제공하는 차세대 교환기 시스템을 실현하기 위해서는 고도의 호처리 능력과 융이한 소프트웨어의 개발이 주목되어 오고 있다. 이러한 차세대 교환기 시스템의 개발에 따르는 전반적인 호처리 특성 및 이를 실현하기 위한 새로운 호처리 프로세서의 구조적인 관점에 대해서 살펴 보기로 한다.

3.1. 호처리 특징

일반적인 컴퓨터의 정보처리와는 달리 호처리에 대해서는 다음과 같은 특징을 갖고 있다 (3).

(1) 실시간 다중처리

다수의 단말로부터 발생되는 서비스 요구에 대해서 각종 톤의 송출, 통화로 형성등은 실시간내에 처리를 해야한다.

(2) 공통자원 관리

각 호에 대해서는 통화로등과 같은 공통자원을 서로 배타적으로 사용하지 않고서 처리가 가능해야 한다.

(3) 이력 의존 처리

각 호에 대한 처리는입력에 따라서 출력이 결정되는 함수형 처리 이외에 입력과 시스템의 내부 상태에 따라서 출력이 결정되므로 이전의 상태를 항상가지고 있으면서 처리가 되어야 한다.

이와같이 특징을 갖는 호처리 프로세서를 데이타 흐름 제어방식을 사용하는 경우에는 다수의 처리소자를 갖는 병렬 프로세서로 호처리 프로세서가 구성이되므로 실시간 다중처리가 가능하며, 데이타 흐름 모델의 토큰 부합방법을 사용해서 공통자원의 관리가 가능하다. 또한 데이타 흐름 시스템을 구현하는데 있어서 언어적 특성으로써 함수형 언어를 사용한다는 점에서 이력 의존 처리의 특징을 구현할 수 있다.

3.2. 시스템 구조

교환기 시스템의 전체적인 구조는 그림3-1과 같이 교환망, 호처리 프로세서 및 단말 인터페이스 구성이 될 수 있다 (2). 여기서 교환망은 실제적으로 호에 대해서 발신자와 착신자를 스위칭하는 TSW (Time Switch)로 구성이 되어 있고, 단말 인터페이스는 외부 장치들에 의해서 발생되는 사건들을 관리하는 부분이다.

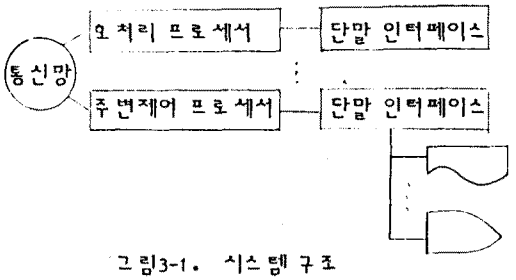
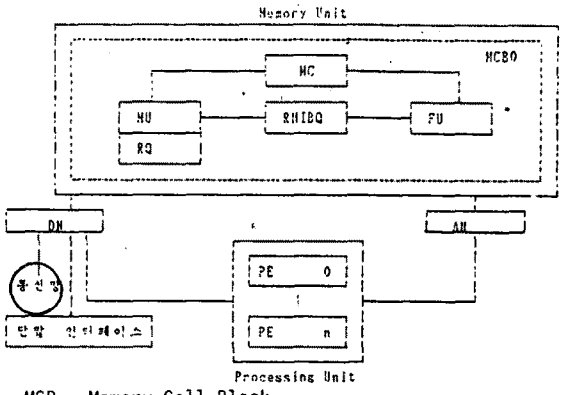


그림3-1. 시스템 구조

호처리 프로세서는 단말 인터페이스에 의해서 수집된 모든 사건을 처리하는 부분으로써 교환기 시스템의 핵심이 되는 호처리 프로그램을 수행한다.

3.3. 호처리 프로세서

호처리 프로세서를 데이타 흐름 시스템으로 구성하는 경우에 일반적인 데이타 흐름 시스템(FORM 1, Manchester Machine, Id 등)에서 갖는 처리 장치, 기억장치 및 Interconnection Network으로 구성이 가능하다 (그림 3-2). 호처리 프로세서의 Interconnection Network으로는 수행이 가능한 연산패킷을 적당한 PE에 넘어주는 Arbitration Network와 PE가 수행한 결과값과 그 결과값을 수행할 피연산자로 써 갖는 Destination Part 로 구성된 데이타 패킷을 기억장치, 단말 인터페이스 또는 통신망으로 보내주는 Distribution Network 이 있다.



MCB : Memory Cell Block
DN : Distribution Network
MU : Matching Unit
PE : Processing Element
FU : Fetch Unit
MC : Memory Cell
RQ : Result Queue
AN : Arbitration Network

그림 3-2. 호처리 프로세서 구조

호처리 프로그램에서 필요로 하는 하나의 원시 연산을 마크로 명령이라고 정의하면, 이러한 여러개의 마크로 명령들이 모여진 것을 마크로 명령블록(MIB)으로 정의 할 수 있다. 호처리 프로그램 설계자가 작성한 호처리 데이타 흐름 프로그램을MIB 의 형태로 번역하는 컴파일러가 존재한다면 여기서 제안하는 호처리 프로세서는MIB 단위의 동시성을 구현할 수 있게 된다. 이러한 MIB단위의 동시성에 기초를 한 호처리 프로세서의 각 구성원들의 기능은 다음과 같다.

(1) 기억 장치

호처리 데이타 흐름 프로그램을 MIB로 번역한 내용과 호처리에 필요한 모든 데이타 및 입력값을 저장하는 영역 및 한 MIB의 모든 피연산자가 존재해서 수행이 가능한 MIB의 주소를 갖는 RMIBQ 로서 구성된다.

CDFPA	Call Data Flow Program Area
DA	Data Area
RMIBQ	Ready MIB Queue

기억장치는 여러개의 집합으로써 하나의 MIB는FU, MU, RQ, MG RMIBQ 로써 구성된다.

(2) 처리 장치

수행이 가능한 명령패킷을 FU로 부터 받아서 실제로 MIB를 수행하고 그 결과를 데이타 패킷으로 형성해서 MU로 보내주는 역할을 하는 PE들의 집합으로 구성된다. 호처리 프로세서에서 필요로 하는 명령패킷과 데이타 패킷의 구조는 다음과 같다.

명령패킷 :

(MIB id , 피연산자 필더, Dest. Part)

데이타패킷 :

(Dest. Part, 데이타 값)

모든PE들은 LM(Local Memory)를 갖게 되는데, 이 LM은 FU 로 부터 넘어온 MIB를 저장하고 그MIB 를 수행한 중간결과 및 PE들 간의 통신에 필요한 데이타를 저장한다.

(3) MU (Matching Unit)

처리 장치로 부터 받아들인 데이타패킷 또는 단말 인터페이스에 의해서 받아들인 사건들을 MIB의 피연산자 필더에 넣어주고, 그 MIB의 모든 피연산자가 존재하는지를 판단해서 수행이 가능한 MIB의 주소를 RMIBQ에 넣어준다. Matching하는 방법으로는 크게 연관 기억장치 (Associative Memory)를 사용하는 방법과 하드웨어 해싱을 이용하는 방법이 있다(8).

(4) FU (Fetch Unit)

RMIBQ 에 있는MIB 의 주소를 가지고 기억장치의 CDFPA에 있는 MIB와 DA에 있는 데이타를 꺼내서 (Fetch), 하나의 명령패킷을 형성해서 처리장치에 넣어준다.

4. 호처리 데이타 흐름 프로그램

고환기 시스템의 호처리 프로그램은 다음과 같은 모듈들로 구성이 된다.

- 발신자 분석
- 착신자 분석
- 입력 숫자 분석
- 시스템 상태 분석
- 타스크

이러한 모듈들과 병행해서 외부장치의 사건을 주사하는 단말 인터페이스 제어 모듈이 있다. 타스크 프로그램은 실제적으로 기억장치에 저장된MIB 들로써 호처리에 필요한 전체적인 프로그램의 구조는 그림4-1과 같다.

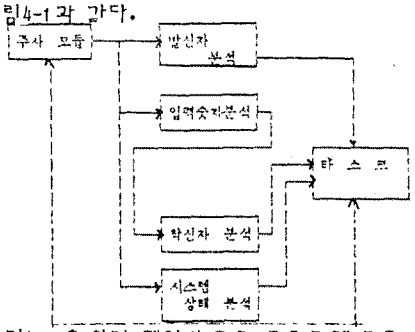
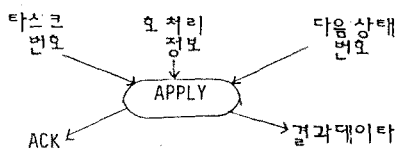


그림4-1.호처리 데이타 흐름 프로그램 구조

타스크 프로그램을 수행시키기 위해서는 타스크번호 호처리 정보및 다음 상태 번호를 가지고서 그 타스크를 호출한다. 타스크를 호출하는 방법은 APPLY Actor를 사용하는데 호처리 데이타 흐름 프로그램에서의 APPLY Actor 의 구조는 다음과 같다.



제안한 데이타 흐름 고환기의 호처리 프로세서가 MIB 단위의 동시성을 제공하기 때문에 APPLY Actor에 의해서 호출되는 타스크는 하나의 MIB가 된다.

5. 결 론

대용량의 고환기 시스템을 실현하는데 있어서Von Neumann 형태의 호처리 프로세서를 구성하는 경우에 있어서는, 이 형태의 고유 제반 문제점으로 인하여 회선증설에 따르는 극한점에 도달하게 된다.

그러나, 앞에서 제안한 데이타 흐름 고환기 시스템의 경우에는 복잡한 프로세서 스케줄링이 없이 간략화된 소프트웨어를 구성할 수 있고, 분산된 여러개의 PE들 간의 서로 독립적으로 수행이 가능하다. 더우기 호처리

프로세서내의 호처리 프로그램을 실행하는 PE를 추가하든지 또는 단말 인터페이스와 관계되는 주사 모듈을 실행하는 PE를 추가함으로써 회선증설에 따른 실시간 처리를 보장할 수 있다. 아울러, PE들간에 호처리 프로그램이 동시에 병렬처리가 가능하므로 시스템의 전반적인 성능향상을 꾀할 수 있다.

본 논문에서 제안한 데이타 흐름 교환기 시스템을 구현하기 위해서는 이 시스템에 적합한 데이타 흐름 프로그램 망 언어를 설계하고, 동시성의 단위인MIB의 구체적인 정의 및 MIB내의 마크로 명령들의 명확한 정의가 필요하며, 아울러 처리장치와 기억장치간의 통신을 위한 Interconnection Network 의 위상 및 구조를 명확히 하는것이 필요하다. 그리고, 시스템의 전체적인 정보를 관리하기 위한 호처리 정보 구조를 데이타 흐름 제어에 적합한 구조로 형성하는 것도 필요하다.

참고 문헌

(1) K.E Matersteck. "No. 4ESS: Prologuw.", Bell System Technical Journal, Vol. 60, No. 6, PART @, July-Aug, 1981, PP. 1041-1048

(2) Digital Exchange for a Wide Range of Applications : Electrical Commu., 1980, Supplement to Vol. 55, No. 2, PP. 10-15

(3) 山田茂樹, 二神新, "呼処理 の 並列性を生かした データフロー 制御交換システム," 電子通信学会論文誌, '86/8, Vol.J69-B, No.8, PP.755-765

(4) M. Ariyama, M. Yamashita, and T. Misu, "DATAFLEX-1: an Experimental Data-Flow Computer Controlled Electronic Switching System." , ISS'84 Florence, 7-11 May 1984, Session 23B

(5) Arvind, D. E. Culer, "Why Dataflow Architectures." MIT/LCS, CSGM-229-1, Sept., 1983

(6) J.B. Dennis, "Data Flow Supercomputers.", IEEE Comp., Vol. 13, No. 11, 1980, PP. 48-56

(7) A.L. Davis, and R.M. Keller, "Data Flow Program Graphs.", IEEE Comp., Vol. 15, No. 2, Feb., 1982, PP. 26-41

(8) F.J. Burkovski, "Parallel Hashing Hardware for Text Scanning Applicatiosn.", Proc1981 Int'l Conf. on Parallel Processing, 1981, IEEE, PP. 282-286

(9) Arvind, and V.Kathail, "A Multiple Processor Dataflow Machine That Supports Generalized Procedures.", MIT/LCS, CSGM-205-1 Feb., 1981