

데이터 처리 가속기 구조

나중화, 김희천, 류대현, 권창희, 정광호, 신승중,
한세대학교 IT학부
e-mail:jwna@hansei.ac.kr

An architecture for data processing accelerator

JongWhoa Na, HeeChern Kim, DaeHyun Ryu*,
ChangHee Kwon, KwangHo Jung, SeungJung Sin,

*Dept of Computer Engineering,
*Dept. of Info. & Comm. Engineering
HanSei University

요 약

본 논문은 날로 증가하는 데이터 처리 요구를 데이터 처리 전용 칩을 이용하여 데이터베이스, 데이터 마이닝, 또는 전문가 시스템 등과 같이 데이터 비교연산에 시간을 많이 소모하는 응용 소프트웨어의 처리 속도를 최소화 할 수 있는 시스템을 제안한다. 본 시스템은 기존의 숫자처리(numeric processing) 보다는 기호처리(symbolic processing)를 위해서 관계 연산(relation operation) 모듈을 이용하여 입력된 데이터들을 하드웨어 레벨에서 고속으로 처리한다. 본 시스템은 칩으로 설계되어 하드 디스크 레벨에서 시스템을 가속 시킬 수도 있고, IP(Intellectual Property)로 구현되어 SoC(System-on-a-chip)의 한 모듈로서 프로세서 레벨에서 시스템을 가속시킬 수도 있다.

1. 서론

최근 인터넷과 컴퓨터 및 유/무선 네트워크 기술의 발달로 인하여 사용하는 데이터의 양이 급증하고 있는 현실이다. 이로 인하여 최근의 고성능 컴퓨터의 개발 동향도 대용량 데이터 처리 성능이 주요 문제로 제기되어 왔다.[1] 날로 증가하는 방대한 규모의 서버급 컴퓨터의 수요를 조사하면, 과학계산용도의 수요보다는 금융, 무역, 보험, 유통 업체에서 사용하는 상업용(database 및 file server용)의 수요가 4배에 이르는 것으로 조사되었다.[2]

전문가 시스템의 경우에도 제한된 범위에서 사용되던 전문가 시스템에서 벗어나는 경우에는 데이터의 양이 급증하므로 처리 속도가 문제가 되어 왔다. 이런 문제점을 극복하기 위하여 병렬처리(parallel processing) 개념이 도입되기도 하였지만

미미한 결과만을 도출하였다. 특히 상식(common sense)을 지식 베이스(knowledge base)로 활용하려는 CYC[ref] 프로젝트는 이제 관련된 제품들이 출시되기 시작하였으며 조만간 상식을 해결할 수 있는 전문가 시스템이 탄생할 것으로 예측된다.[3]

이렇게 된 이유는 구조적으로 컴퓨터가 노이만(Neumann) 구조를 가지므로 모든 코드와 데이터가 숫자 처리(Numerical processing)기능을 중심으로 설계 및 운용되어왔기 때문이다.[ref] 본 논문에서는 데이터 베이스 혹은 전문가 시스템에서 날로 증가하는 데이터 처리 작업의 성능향상을 도모할 수 있는 데이터 처리 하드웨어 가속기 구조와 시뮬레이션 모델을 제안한다.

2. 데이터 처리 가속기의 개요

데이터처리가속기는 하드 디스크 등의 저장장치와 컴퓨터사이에 위치해서 전송할 데이터의 양을 지능적으로 축소 시켜서 컴퓨터에 전달하여 응용 소프트웨어의 실행속도를 향상시키는 장치이다. 먼저 데이터베이스 분야에서 가속기의 응용 예를 먼저 살펴보자.

데이터베이스 응용분야에서의 데이터 처리 가속기의 기본개념은 다음과 같다. 즉 관계형 데이터베이스 연산들 중에서 가장 빈번하게 사용되는 그래서 총 처리시간에서 가장 많은 계산시간을 소모하는 연산을 소프트웨어 레벨이 아닌 하드웨어 레벨에서 전처리(pre-processing)함으로써 성능 개선하는 것을 목표로 한다. 본 논문에서는 관계형 연산들 중에서 기본적인 선택(selection)연산과 투사(projection)연산을 우선 구현하고 나머지 주요 연산은 계속적으로 연구할 계획이다.

데이터 처리 가속기는 가속기 제어 소프트웨어와 데이터 처리 가속기를 탑재한 하드 디스크로 구성된다. 가속기 제어 소프트웨어는 사용자의 질의어를 분석하여 가속기를 사용할 것인지의 여부를 판단한다. 만일 가속기를 사용할 수 있는 것으로 판단되면, 가속기가 연산을 수행할 수 있는 상태로 초기화하고 대기하다가 하드 디스크로부터 데이터가 입력되면 사용자의 질의를 처리한다. 가속기를 초기화는 다음 섹션에 설명된다.

데이터 처리 가속기의 역할은 이때 입력되는 데이터의 양을 최소한으로 만드는 역할을 수행하여 전체적인 처리 시간을 감소시킨다. 즉 질의어를 처리하는 목적에 필요 없는 연산들은 미리 하드 디스크 레벨에서 폐기함으로써 실제로 컴퓨터까지 전달되는 데이터의 양을 최소화하여 전체적인 질의의 응답시간을 최소화 하는 것을 목적으로 한다. 그림 1에는 은행의 고객 입출금내역을 기록하는 데이터베이스(a)를 활용한 선택연산 및 투사연산의 결과가 그림 (b)와 (c)에 각각 설명되어 있다.

같은 맥락에서 전문가 시스템(Expert System)도 데이터 처리 가속기를 이용하여 성능을 향상시키거나 혹은 실행시간의 문제로 인하여 도외시되었던 응용 분야를 새롭게 접근할 수 있다. 전문가 시스템에서는 if-then-else의 규칙들과 프레임(frame)등의 사실(fact)들로 해결하고자 하는 문제들을 표현하게 된다. 전문가 시스템이 일반적인 사실들을 표현하려면 수많은 규칙들과 그 규칙들이 활용해야 하는 사실들이 지식베이스(knowledge base)

에 존재해야 한다. 그런데 전문가 시스템에서 문제를 해결하려면 지식베이스 내의 수많은 규칙들의 조건(condition)들이 더 많은 수의 사실(fact)들과 비교해야 하기 때문에 비교 연산(match operation)이 매우 시간을 많이 소모하는 것으로 알려져 있다.

이 비교연산은 전문가 시스템에서 지원하는 규칙과 규칙의 조건들, 그리고 프레임 등으로 표현되는 변수의 개수가 많을수록 긴 연산시간이 소요된다. 이는 전문가 시스템의 응용 범위 혹은 전문성의 범위가 넓어질수록 관련된 변수가 많아지므로 소요시간도 함께 증가한다. 따라서 소프트웨어로 구현하는 전문가 시스템을 데이터 처리 가속기를 이용하면 성능향상을 도모할 수 있게 된다.

TransNo	CustomerNo	TimeStamp	Amount	Type
0014744	11224	199905151511	100,000	입금
0123443	22234	199905180543	3,000,000	입금
3453566	34342	199905181645	45,000,000	출금
4454334	22234	199905260955	2,500,000	입금
6535356	67876	199906031130	120,000	출금
...

(a) 은행업무 데이터베이스 예

• Projection on TransNo

TransNo
0014744
0123443
3453566
4454334
6535356

(b) TransNo를 이용한 프로젝션 연산의 결과

• Selection w/CustNo=22234

0123443	22234	199905180543	...
4454334	22234	199905260955	...

(c) CustNo를 이용한 selection 연산 결과

그림 1 데이터 필터 연산의 예.

3. 데이터 처리 가속기 구조

일반적으로 IDE 또는 ATA방식의 하드 디스크의 운용은 다음과 같다. 이전의 디스크는 플럭스(flux)의 천이(reversal)를 검출하여 '1'과 '0'을 판독했으나 IDE/ATA방식에서는 EPRML (Extended Partial Response Maximum Likelihood) 방식을 사용하여 플래터에서부터 디스크 암(Arm)에서 읽어오는 아나로그 데이터 스트림을 디지털 신호 샘플링하고 처리한 뒤, 검출 알고리즘을 이용하여 디지털 데이터 '1'과 '0'을 판독하고, 이 디지털 데이터를 RLL (Run Length Limited) 방식으로 인코딩/디코딩(encoding/decoding)하여 사용한다. 디스크 암(Arm)에서 읽혀진 데이터는 버퍼에 저장되었다가 메모리로 전송된다.[4,5]

데이터 처리 가속기(Data Processing Accelerator)를 내장한 하드 디스크 드라이브의 전체 블록도가 그림 2에 도시되어 있다. 그림 2의 디스크의 플래터(platter)에서 읽혀서 버퍼에 저장된 데이터는 디스크의 가속기 사용여부에 따라서 수행하는 연산이 달라진다.

먼저 가속기를 사용하는 경우에는 데이터 처리 가속기 모듈로 데이터를 전송하여 지정된 연산을 수행하면 된다. 만일 가속기를 사용하지 않는 경우에는 일반적인 경우에서처럼 버퍼에서 외부 인터페이스로 직접 전송한다.

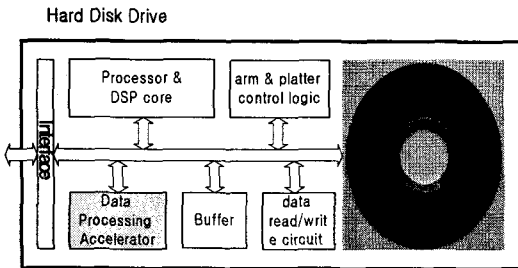


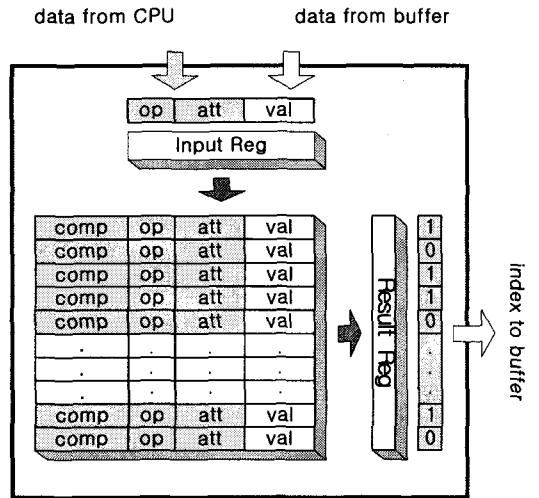
그림 2 데이터 처리 가속기를 활용한 하드 디스크 드라이브 블록도.

그림 3에서는 데이터 처리 가속기의 상세하게 설명한다. 사용자의 질의를 입력받은 가속기 제어 소프트웨어는 질의에 필요한 연산과 어트리뷰트(attribute) 값을 추출하여 데이터 처리 가속기가 사용할 3 비트 명령코드인 op 값과 32비트 어트리뷰트의 값을 att 로 변환한다. 그 다음에는 op 와 att 를 하드 디스크 드라이브 내의 데이터 처리 가속기로 전달하고 대기상태로 들어간다. 가속기 제어 소

프트웨어는 하드디스크에서 선택된 데이터를 데이터베이스로 전달하고 작업을 종료하게 된다.

하드 디스크내의 데이터 처리 가속기는 그림 3의 comp, op, att 부분의 값은 시스템으로부터 입력받고 val 부분은 버퍼 메모리에서 입력 받는다. op, att, 그리고 val 값은 임시로 Input 레지스터에 저장된다. comp 모듈은 32비트 att와 32비트 val을 입력받아서 32비트 비교연산을 수행한다. 비교연산은 관계형 데이터베이스에서 활용하는 =, ≠, <, ≤, >, ≥ 의 여섯 가지 연산을 수행해야 하므로 비교기(comparator)를 활용하였다. op는 6가지 기본 연산과 가속기의 comparator를 초기화 하는 제어 명령을 포함할 수 있도록 3 비트를 할당한다. att 및 val 레지스터는 1차 버전에서는 각각 32비트로 설정하였다. 비교연산의 결과는 Result 레지스터에 저장되며 Result 레지스터는 각 행마다 1 비트를 할당하여 선택의 여부를 나타내도록 구성하였다.

Data Processing Accelerator Module



Legend

- comp** 32bit Comparator
수행연산: <, ≤, ≠, >, ≥
- op** 3bit relation operation code
수행할 연산 선택명령
- att** 32bit variable for attribute from CPU
- val** 32bit value from buffer

그림 3 데이터 처리 가속기 블록도.

그림 3의 Result 레지스터의 값은 버퍼메모리의 각 열(row)의 인덱스로 활용하여 '1'인 열의 데이터는 메모리로 전송되고 '0'인 열의 데이터는 폐기한다.

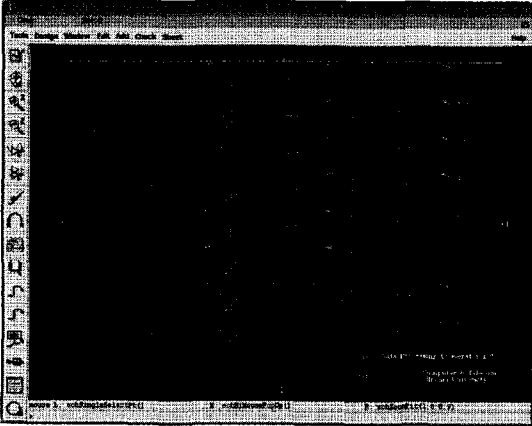


그림 4 데이터 처리 가속기의 schematic diagram.

그림 4에는 데이터 처리 가속기를 SUN ultra60에서 cadence를 이용하여 schematic으로 구현한 결과이다. 여기서는 SIFT_CELL은 버퍼로부터의 32-bit 데이터와 시스템으로부터의 32-bit 데이터가 FIFO의 형태로 구성되어 있다. 8 clock 이후에 8개의 SIFT_CELL이 모두 load 되면, 시스템의 op를 이용하여 COMP_CELL은 op가 지정하는 연산 즉, =, ≠, >, ≥, <, ≤ 연산 중의 한 연산을 수행한다. 연산 결과 값은 8개의 1-bit Register에 각각 저장하게 된다.

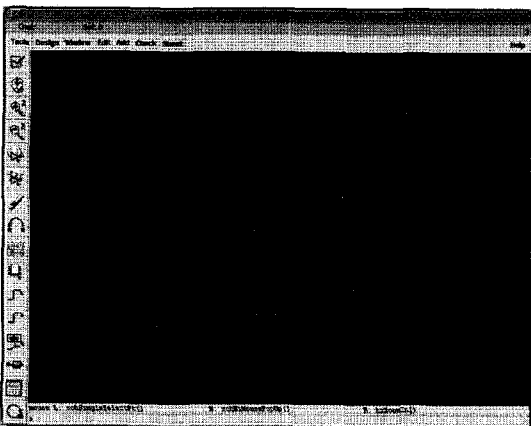


그림 5 데이터 처리 가속기의 세 모듈의 schmetic diagram.

그림 5는 데이터 처리 가속기의 주요 모듈인 SIFT_CELL과 COMP_CELL, 그리고 REG_1BIT 모듈의 상세 블록도가 도시되어 있다.

4. 결론 및 향후 연구

본 논문에서는 대용량 데이터를 처리해야 하는 응용 분야에 적합한 데이터 처리 가속기의 구조를 제안하였다. 제안된 가속기는 Cadence 사의 CAD 도구를 이용하여 기능을 확인하였다. 가속기는 IP의 형태로 디스크 드라이브 제어용 SoC에서 활용될 수 있으므로 제품의 저가격화, 저전력 소모, 신뢰성 향상등에 기여할 것으로 예상할 수 있다.

하드 디스크 드라이브는 프로세서나 반도체 메모리보다는 덜 주목을 받았으나 상대적으로 기술개발이 덜 주목을 받았다. 그간의 연구도 재료 및 회로 분야로 한정되어 있어왔고 시스템 레벨에서는 버퍼 또는 캐쉬 메모리가 장착되었지만 IDE/ATA 제품의 경우에는 버퍼 용량이 2M 바이트, 그리고 SCSI의 디스크의 경우에도 16M 바이트 정도로 하드 100G를 넘는 디스크 용량에 비해 별다른 디스크의 성능개선 연구가 진행되지 못해왔다. 그러나 대용량 데이터 처리 작업들은 대개 하드 디스크에 데이터를 저장하므로 하드디스크 레벨에서 빠른 데이터 처리는 시스템 측면에서 매우 중요한 역할을 차지한다.

본 칩의 성능은 가속기를 탑재한 디스크가 포함된 컴퓨터에서 실제 프로그램을 실행시켜서 데이터를 추출해서 성능 평가를 할 계획이다. 데이터 처리 가속기 칩은 우선 FPGA로 구현하여 하드 드라이브 제어회로에 연결하여 시험용 디스크를 구성할 계획이다.

참고문헌

- [1] Roger S. Pressman "Software Engineering A Practitiners' Approach" 3rd Ed. McGraw Hill.
- [2] P. Stenstrom, et al., "Trends in shared memory multiprocessing", IEEE Computer, pp. 44-50, Dec. 1997.
- [3] <http://www.cvc.com> /
- [4] http://storagereview.com/guide/guide_index.html
- [5] <http://www.maxtor.com/en/technologies/index.htm>, technology overview.