확장 가능한 범용 Associative Processor 구조 및 응용

윤재복, 김주영, 김진욱, 박태근
가톨릭대학교 정보통신전자공학부
e-mail : qjrl77@konet.net, mog016@catholic.ac.kr
white3479@hanmail.net, parktg@catholic.ac.kr

Architecture of a scalable general-purpose associative processor and its applications

Jae-Bok Yun, Ju-Young Kim, Jin-Wook Kim, Tae-Geun Park
School of Information, Communication, and Electronic Engineering
The Catholic University of Korea

Abstract

일반 컴퓨터에서 중앙처리장치와 메모리 사이의 병목현상인 "Von Neumann Bottleneck"을 보이는데 본 논문에서는 이러한 문제점을 해소하고 검색위주의 응용분야에서 우수한 성능을 보이는 확장 가능한 범용 Associative Processor(AP) 구조를 제안하였다. 본 연구에서는 Associative computing을 효율적으로 수행할 수 있는 명령어 세트를 제안하였으며 다양하고 대용량 응용분야에도 적용할 수 있도록 구조를 확장 가능하게 설계함으로써 유연한 구조를 갖는다. 12 가지의 명령어가 정의되었으며 프로그램이 효율적으로 수행될 수 있도록 명령어 셋을 구성하고 연속된 명령어를 하나의 명령어로 구현함으로써 처리시간을 단축하였다. 제안된 프로세서는 bit-serial, word-parallel로 동작하며 대용량 병렬 SIMD 구조를 갖는 32 비트 범용 병렬 프로세서로 동작한다. 포괄적인 검증을 위하여 명령어 단위의 검증 뿐 아니라 최대/최소 검색, 이상/이하 검색, 병렬 덧셈 등의 기본적인 병렬 알고리즘을 검증하였으며 알고리즘은 처리 데이터의 개수와는 무관한 상수의 복잡도 O(k)를 갖으며 데이터의 비트 수만큼의 이터레이션을 갖는다.

I. 서론

일반적인 개념에서 범용 컴퓨터는 중앙처리장치(CPU)와 메모리로 구성되는 Von Neumann 구조를 중심으로 발전하여 왔다. 최근 컴퓨터 및 반도체 기술의 발전으로 메모리의 용량은 수 백 메가바이트 범위에 이르고, 단일 프로세서 구조의 경우에 하나의 CPU가 많은 양의 메모리를 한번에 하나씩 읽어 처리해야 하기 때문에 CPU 자체의 처리성능과 무관하게 전체 시스템의 성능이 저하될 수 있다.

검색위주의 응용분야에서 Content-Addressable Memory(CAM)는 효과적인 해답이 될 수 있다. 메모리에 저장된 데이터에 접근할 때에 주소를 이용하여 데이터에 접근하는 RAM과는 달리 CAM은 저장된 내용을 검색함으로써 데이터에 접근하고 처리한다. 이러한 CAM을 기초로 하여 각 워드에 단순한 Processing Element(PE)를 첨가하면 SIMD 형태의 병렬 Associative Processor(AP)를 구성할 수 있다. 따라서 이 구조는 수 백~수 천 개의 PE들이 네트워크로 연결된 SIMD 구조로 이해될 수 있으며 이때 내부에 설계되는 콘트롤러는 전체 시스템의 수행을 제어하게 된다. 또한 다양한 응용 분야에 적용되기 위하여 효율적인 명령어 구조가 제안되어야 하며 제안된 구조는 보조 프로세서로서 호스트와 연결되는 일반적인 SIMD 시스템과 같이 사용될 수 있다[1].

본 연구에서는 Associative computing을 기반으로 하는 효율적인 명령어 세트를 제안하였다. 또한 다양하고 대용량 응용분야에 적용할 수 있도록 구조 확장이 가능하게 설계함으로써 유연한 구조를 갖는다. 전체 시스템

구조를 정의된 명령어 세트가 수행되기에 적합하도록 구성하고 이를 설계하여 검증하였다. 본 논문에서 제안하는 AP는 그 구조가 상대적으로 단순하지만 효율적이고 유연한 구조를 갖고 있으며 하나의 칩에 수 천 개의 PE를 내장할 수 있다.

본 논문의 제 2장에서는 기존에 제안되었던 AP들의 구조와 특징을 분석하고 설명한다. 제 3장에서는 제안된 확장성 있는 범용 AP 구조 및 명령어 등에 대하여 설명하고, 제 4장에서는 설계 및 모의실험을 분석하고 마지막으로 제 5장에서는 본 논문의 결론을 제시한다.

Ⅱ. Associative Processor

현재까지 Associative Computing을 이용하여 데이터베이스, 인공지능, 패턴인식, 전문가시스템, 영상처리, 네트워크 등 여러 가지 응용분야에서 많은 연구가 진행되어 왔다[2-4]. Storer 등은 이종 비젼 아키텍처(Heterogeneous Vision Architecture)를 위한 AP 시스템을 제안하였다[5]. GLiTCH라고 불리는 AP 칩은 8 비트 영상을 처리할 수 있도록 64 비트 CAM을 내장하고 있으며 명령어를 저장하는 ROM과 단순한 1 비트 ALU를 갖는 PE 그리고 단순한 네트워크와 제어 회로로 구성되어 있다. Higuchi 등은 인공지능에 적합한 IXM2 병렬 AP 시스템을 제안하였다[6]. 제안된 시스템은 자연어처리 분야를 주 응용분야로 설계되었다. rule-based 방법은 수행속도, 확장성, 질, 규칙의 정의 등의 면에서 많은 문제를 지니고 있지만 CAM 메모리를 기본으로 하는 대용량 병렬 SIMD 방법은 언어번역과 같은 실시간 응용분야에 적합하다. Louri 등은 방대한 데이터베이스를 고속으로 처리하기 위하여 Optical associative computing을 적용한 시스템을 제안하였다[7].

특정한 응용분야에 적합하도록 제안된 구조들 외에 좀 더 다양한 응용분야에 적용될 수 있는 범용의 구조들도 여러 가지 제안되었다. Stormon 등이 제안한 AP 시스템은 비교적 단순한 구조를 갖지만 유연한 구조 때문에 그의 집적도가 높고 따라서 상대적으로 큰 응용문제를 다룰 수 있다는 장점이 있다[10]. 최근에 제안된 bit-parallel block-parallel AP는 입력 벡터와 모든 검색 대상의 벡터 간에 병렬로 연산되며 비트뿐 아니라 블록 간에도 병렬로 처리가 가능함으로써 성능이 개선되었다[11]. Grosspietsch 등은 CAPRA라는 AP 시스템을 제안하였는데 일반적으로 데이터베이스 검색, 단순한 수리연산, 영상처리 등에 적용할 수 있으며 큰 특징은 구조의 유연성과 테스트기능(testability)과 오류내성기능(fault tolerance)으로 요약할 수 있다[8]. 일반적인 AP의 PE에 비하여 복잡한 구조를 갖고 있으며 기능 또한 많이 구현되어 있다. 제안된 시스템은 일반적으로 데이터베이스 검색, 비교적 단순한 수리연산, 영상처리(압축, 필터링, 패턴인식, 등) 등에 적용할 수 있다. Tavangarian이 제안한 AP 시스템은 워드단위를 기초로 하는 데이터를 플래그를 기초로 하는 형태로 변환하여 처리한다[9].

Ⅲ. 제안된 Associative Processor 구조

그림 1은 제안된 확장 가능한 범용 병렬 AP의 시스템 구조이다. 데이터 입출력은 32 비트로서 내부에 있는 5종류의 레지스터에 저장된다. 정의된 명령어는 12 가지이며 입력된 명령어는 on-chip controller에서 디코딩되고 실행된다. 오른쪽의 CAM 블록은 42 비트 단위의 N개의 워드로 구성되어 있고 각 워드는 병렬 산술연산 등 다양한 알고리즘 적용을 위하여 비트별 선택기능을 가지고 있다. 상위 10 비트는 응용 시에 데이터의 구분을 위한 태그 비트로 사용된다. 또한 PE 블록은 CAM을 이용한 검색 결과를 이용하여 다양한 query를 처리할 수 있는 단순한 기능의 1 비트 프로세서이다. 이와 같이 구현된 AP 시스템은 문제 크기에 따라 여러 개의 AP 칩들을 직렬 연결하여 확장 가능하도록 설계되었다. 이 때 인접한 AP 칩들 간에 데이터를 주고받기 위하여 최상위와 최하위 PE의 R1 레지스터를 서로 연결하면 쉽게 PE 용량을 확장할 수 있다. 이 때 뒤에서 설명할 MRR 트리 구조도 확장되어야 하는데 이는 보드레벨에서 간단한 로직을 추가하여 구현할 수 있다.

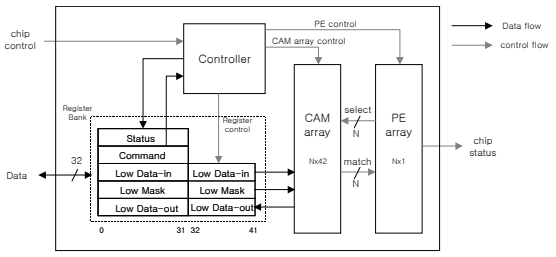

그림 1. 제안된 범용 병렬 Associative Processor 구조

가. CAM 어레이

CAM은 기존의 RAM을 확장한 형태로서 각 셀은 쓰기와 읽기 연산이 가능할 뿐만 아니라 매치연산을 할 수 있다. 아래의 그림 2에서 상위의 인버터 2개와 트랜지스터 2개는 기존의 RAM부분이고, 하위의 트랜지스터 4개가 매치연산을 위한 구조이다. 매치 라인은 매치연산을 수행하기 전에 선 충전(precharge)되며 저장된 값과 외부에서 인가된 값을 비교하여 선택적으로 방전됨으로써 매치를 수행한다. 정적 CAM 셀을 이용하여 하나의 칩으로 만들기에 적당한 1024 워드 x 42 비트 크기의

CAM 어레이를 구성한다. CAM 어레이는 어드레스가 없으며 워드 선택선이 이를 대신한다. 읽기 동작 시에는 하나의 데이터만을 읽을 수 있지만 쓰기 동작 시에는 여러 위치에 같은 데이터를 동시에 쓸 수 있다. 매치 선은 wired-AND로 연결되어 있기 때문에 하나의 비트라도 매치되지 않으면 매치에 실패한 것으로 판명된다. 본 구조에서는 쓰기 명령어 수행 시에도 각각의 비트에 원하는 대로 마스크를 씌울 수 있기 때문에 하나의 워드 중 원하는 일부 비트에만 접근하는 것이 가능하다.

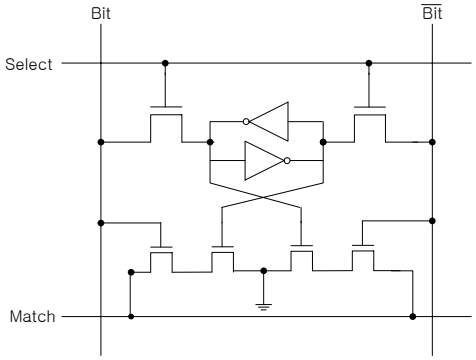

그림 2. static CAM 셀의 구조

나. PE 어레이

각각의 CAM 워드는 데이터를 처리하기 위한 PE와 결합하여 SIMD 구조를 갖는다. 하나의 PE는 매치 결과를 저장하기 위한 세 개의 1 비트 레지스터(R1, R2, R3), 단순한 부울 연산을 할 수 있는 논리 블록(GPLB, general purpose logic block), 상하의 PE와 데이터 통신을 할 수 있는 선형 네트워크 등을 포함하는 Row logic 블록과 두 개 이상의 매치 결과가 나왔을 경우 하나의 결과를 선택할 수 있는 우선순위 인코더(MRR, multiple response resolver) 트리로 구성된다.

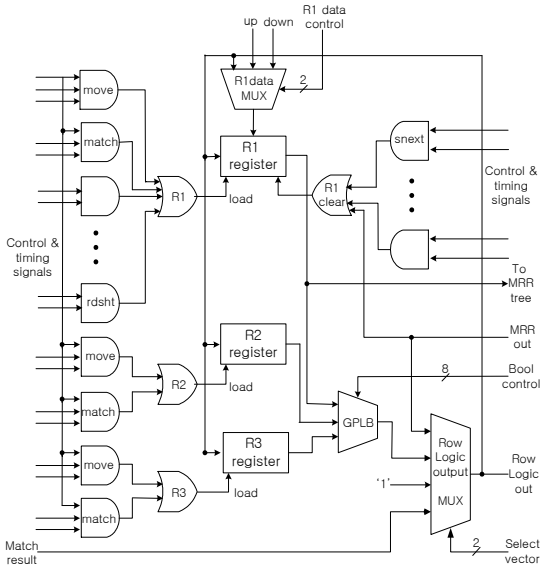

그림 3. Row logic 구조

그림 3은 Row logic 블록의 구조를 나타낸다. 좌측의 신호들은 on-chip controller에서 만들어진 제어신호들을 나타낸다. up, down은 각각 위쪽과 아래쪽 PE의 R1 레지스터를 의미하며 R1, R2, R3 등의 1-bit 레지스터들은 내부 연산 결과들을 저장하기 위해 이용된다. Move와 Match 연산 시 발생하는 결과 값은 3 개의 레지스터 어디에나 저장될 수 있으나 데이터를 쉬프트하거나 MRR 트리에 입력으로 들어갈 수 있는 기능은 R1 레지스터만 가능하다. GPLB는 부울 연산을 담당하는 블록으로 R1, R2, R3 3개의 레지스터 값을 입력으로 8 가지 minterm을 이용하여 256가지 함수의 부울 연산을 한다. PE 출력단에 있는 멀티플렉서는 MRR 결과, GPLB 결과, Match 결과, all '1' 중의 하나를 선택하여 PE의 출력 값으로 내보낸다. Row_logic_out은 CAM 어레이의 워드 선택선으로 사용되어 읽기나 쓰기 동작 시에 해당 워드를 선택한다.

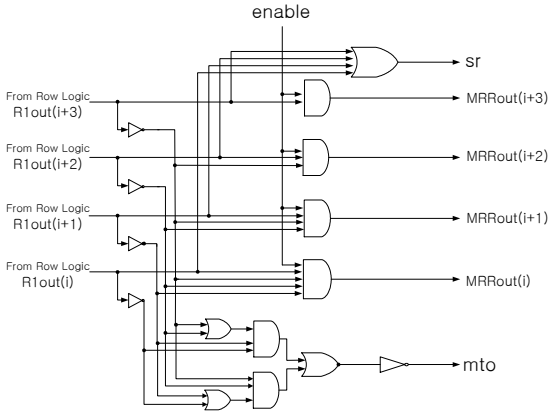

그림 4. 하나의 MRR 블록 구조

CAM은 그 구조적 특성상 동시에 여러 개의 워드에 쓰기 혹은 매치 연산을 실행할 수는 있으나, 읽기 연산의 수행 시에는 한 번에 단 하나의 워드만을 읽어낼 수 있다. 즉 여러 개의 1을 가진 응답벡터(response vector)에 해당하는 워드들을 읽을 경우, 복수의 응답벡터 중에서 최상위에 "1"로 세팅된 것을 선택할 수 있는 우선순위 인코더를 필요로 한다. 이로써 생성된 하나의 응답 출력을 선택선으로 하여 읽기를 수행한다. 이를 담당하는 우선순위 인코더가 MRR이며 이는 다단계 트리 구조로 되어 있다. 그림4는 인접해 있는 네 개의 PE로부터 R1 입력을 받아 처리하는 하나의 MRR 블록 구조를 나타내며 1024개의 PE를 처리하기 위해서는 MRR 트리 구조를 사용한다. MRR 트리는 여러 개의 "1"을 가진 입력이 들어올 경우 최상위의 "1"만 남겨두고 나머지는 "0"으로 출력한다. 그 다음 워드를 읽기 위해선 응답벡터의 최상위의 R1 값을 리셋하면 차상위의 "1"을 가진 워드를 선택할 수 있다. MRR은 출력 응답벡터를 만드는 동시에

SR(some response)와 MTO(more than one) 신호도 함께 생성한다. SR은 응답벡터 중에 “1” 이 하나 이상 있는가에 대해 판단해주며, MTO는 응답벡터 중에 “1”이 2개 이상 있는가를 판단한다.

Ⅳ. 설계 및 검증

본 논문에서 제안된 범용 AP는 VHDL로 설계되어 Modelsim 환경에서 기능이 검증되었다. 좀 더 포괄적인 검증을 위하여 명령어 단위의 검증 뿐 아니라 다음의 병렬 알고리즘의 수행을 검증하였다. 최대값/최소값 검색, 이상/이하 검색(Greater-than/Less-than search), 병렬 가감산 등의 알고리즘도 검증하였다. 모든 알고리즘은 병렬로 수행되기 때문에 저장된 데이터의 개수와는 무관하며 상수 O(k)의 복잡도를 가지며 처리 비트 수만큼의 이터레이션이 필요하다.

가. 최대값 검색 알고리즘

CAM에 저장된 데이터 중 가장 큰 값을 찾아내는 알고리즘이다. MSB부터 LSB까지 이터레이션마다 한 비트씩 ‘1’과 매치를 수행하면서 반응벡터의 결과 값을 이전 결과와 AND 연산하여 저장한다. 반응벡터가 단 하나의 1을 갖거나 LSB까지 검색 완료될 때까지 반복한다. 만일 AND 연산 후에 반응벡터가 모두 ‘0’이면 해당 비트로는 경우의 수를 줄일 수 없다는 뜻이므로 원래의 반응벡터 값을 복원하여 다음 비트에 대하여 검색을 진행한다. 마지막까지 검색한 후 남는 것이 최대값이며 반응벡터 중에 ‘1’인 엔트리가 두 개 이상 남는다면 모두 같은 값이다.

나. 병렬 덧셈 알고리즘

CAM에 저장된 데이터를 기반으로 병렬 덧셈을 수행할 수 있다. CAM의 32비트 데이터 영역에 상위 8 비트(24-31)에는 A 값이 다음 8 비트(16-23)에는 B 값이 저장되어 있고 이 두 값을 더한 결과는 하위 16 비트(0-15)에 C 값으로 저장된다. LSB부터 시작하여 ai와 bi 비트는 match 연산에 의해 해당 PE에 저장되고 GPLB의 연산에 의해 carry와 sum이 계산되어 CAM의 C 값의 해당 비트에 저장된다. 덧셈 연산은 bit-serial, word-parallel의 형태로 동작되어 A, B 데이터의 개수에 상관없이 병렬로 수행된다.

Ⅴ. 결 론

본 논문에서는 Associative computing을 기반으로 하는 범용 병렬 AP 프로세서 구조를 제안하고 다양하고 대용량 응용분야에도 적용 가능하도록 구조를 확장 가능하게 설계함으로써 제안된 프로세서는 bit-serial, word-parallel로 동작하는 대용량 병렬 SIMD 프로세서이다. 설계된 PE는 Response Registers, 부울 함수 연산기(GPLB), MRR(Multiple Response Resolver) 트리, 쉬프트 레지스터 등을 포함하는 1 비트 구조로 복잡도를 개선하였으며 추가의 하드웨어 없이 Binary 모드뿐만 아니라 영상처리, 패턴인식, 신경망시스템 등 다양한 분야에서 필요한 Quad 모드(0, 1, *, N)도 지원할 수 있도록 설계되었다. 제안된 구조를 검증하기 위하여 제안된 명령어뿐만 아니라 기본적인 검색 및 연산 알고리즘을 구현하여 실험하였다.

감사의 글

저자들은 본 연구를 위하여 설계 환경을 제공하여 준 IDEC(IC Design Education Center)에 감사드린다.

참고문헌

[1] J. Potter, Associative computing: A programming paradigm for massively parallel computers, Plenum publishing, New York, 1992.

[2] A. Krikelis, "Computer vision applications with the associative string processor," J. of parallel and distributed computing, vol.13, no.2, pp.170-184, 1991.

[3] H. Kitano, Speech-to-speech translation: A massively parallel memory-based approach, Kluwer academic publisher, 1994

[4] C. Weems, "The image understanding architecture," Int'l J. computer vision, vol.2, no.4, pp.251-282, 1989.

[5] R. Storer, et al., "An associative processing module for a heterogeneous vision architecture," IEEE Micro, vol.12, no.3, pp.42-55, 1992.

[6] T. Higuchi, et al., "The IXM2 parallel associative processor for AI," IEEE Computer, vol.27, no.11, pp.53-63, 1994.

[7] A. Louri, et al., "An optical associative parallel processor for high-speed database processing," IEEE Computer, vol.27, no.11, pp.65-72, 1994.

[8] K. Grosspietsch and R. Reetz, "The associative processor system CAPRA: architecture and applications," IEEE Micro, vol.12, no.3, pp.58-67, 1992.

[9] D. Tavangarian, "Flag-oriented parallel associative architectures and applications," IEEE Computer, vol.27, no.11, pp.41-51, 1994.

[10] C. Stormon, et al., "A general-purpose CMOS associative processor IC and system," IEEE Micro, vol.12, no.3, pp.68-78, 1992.

[11] K. Tamaru, K. Kobayashi, and H. Onodera, "Memory based architecture and its implementation scheme named bit-parallel block-parallel functional memory type parallel processor MPMP FMPP," Comp. and Elect. Engineering, vol.24, pp.17-31, 1998.