

# 고속의 인터넷 라우터를 위한 IP 룩업구조 설계

준회원 서해준\*, 정회원 안희일\*\*, 조태원\*

## A Design of the IP Lookup Architecture for High-Speed Internet Router

Hae-Jun Seo\* *Associated Member*

Hee-Il Ahn\*\*, *Regular Members*, Tae-won Cho\* *Regular Members*

### 요 약

고속 라우터의 인터넷 패킷 처리에서 가장 많은 시간이 걸리는 부분이 IP 패킷 주소 룩업 중 LPM 탐색이다. 기존의 CAM을 이용한 LPM 탐색에서 LPM 탐색율이 높으면서 동시에 복잡도도 높지 않은 방식은 룩업 테이블의 갱신시간이  $O(n)$ 으로 오래 걸렸다. 본 논문에서 설계한 파이프라인 룩업 테이블은 고속 LPM 탐색을 위한 구조로서 갱신시간이  $O(1)$ 으로 짧으면서도, LPM 탐색율이 높고, 복잡도도 높지 않은 새로운 방식의 파이프라인 구조로, 1bit RAM 셀을 이용한 CAM 배열 구조로 설계하였다. 룩업 테이블은 3단계의 파이프라인으로 구성된다. 단계1 및 단계2의 키 필드 분할 수 및 매칭점의 분포에 따라 파이프라인의 성능이 좌우되며, LPM 탐색율이 달라질 수 있다. 설계방식은 RTL에서 하드웨어 기술 언어를 이용해서 수행되었고, 0.35 $\mu$ m CMOS 표준 셀 라이브러리를 이용해서 게이트 수준에서 기능을 검증하였다.

Key Words : Router; IP address; Lookup table; CAM; LPM.

### ABSTRACT

LPM(Longest Prefix Matching)searching in IP address lookup is a major bottleneck of IP packet processing in the high speed router. In the conventional lookup table for the LPM searching in CAM(Content Addressable Memory) the complexity of fast update take  $O(1)$ . In this paper, we designed pipeline architecture for fast update of  $O(1)$  cycle of lookup table and high throughput and low area complexity on LPM searching. Lookup-table architecture was designed by CAM(Content Addressable Memory)array that uses 1bit RAM(Random Access Memory)cell. It has three pipeline stages. Its LPM searching rate is affected by both the number of key field blocks in stage 1 and stage 2, and distribution of matching point. The RTL(Register Transistor Level) design is carried out using Verilog-HDL. The functional verification is thoroughly done at the gate level using 0.35 $\mu$ m CMOS SEC standard cell library.

\* 충북대학교 전기전자 및 컴퓨터공학부 디지털시스템 연구실(hjseo@dssd.chungbuk.ac.kr, twcho@cbucc.chungbuk.ac.kr)

\*\* 한국조폐공사 기술연구소 정보기술연구팀(hiahn@komsep.com)

논문번호 : 030213-0519, 접수일자 : 2003년 7월 12일

\* 본 연구는 대전 IDEC과 정보통신연구진흥원 정보통신기초기술 연구지원사업 지원으로 수행되었습니다

## I. 서 론

차세대 IP(Internet Protocol) 라우터(router) 개발에 있어서 가장 중요하고 시급한 설계의 문제점은 IP 경로 탐색 메커니즘이다. IP 주소 룩업(lookup)은 고속 인터넷 라우터의 성능을 좌우하는 주요 요소이며, 라우터의 인터넷 패킷(internet packet)처리에서 가장 많은 시간이 걸리는 부분이 IP 패킷 주소 룩업 중 LPM(Longest Prefix Matching)탐색이다. 인터넷 사용자의 증가와 사용 대역폭의 증가는 더욱더 높은 트래픽(traffic)을 유발시켜, 네트워크(network)를 연결하는 라우터의 고속화가 절실하게 요구되었다. 이러한 라우터의 고속화의 핵심은 IP의 처리를 어떻게 고속화하는가에 전적으로 의존된다. 라우터를 고속화하기 위해서는 3계층(layer3)의 IP 전송부분을 고속화시키고, 하드웨어에 의한 고속의 패킷 연결 변환 장치(packet switching)를 사용하여 다중 병렬연결 구조를 가지는 네트워크를 형성하여야 한다.

인터넷 패킷 통신에서 패킷의 목적지 주소에 따라 패킷이 전달될 출력포트(next hop)를 결정하는 것을 IP 주소 룩업(lookup)이라 한다. 라우터는 패킷이 도착하면 도착 패킷의 목적지 IP 주소를 참조하여 룩업 테이블(lookup table)에서 LPM을 수행하여 최종 목적지 출력포트를 찾아낸다. 기존의 LPM탐색 방법은 라우터의 룩업 테이블의 갱신 속도에 대한 고려 없이, LPM 탐색율(throughput)만 높이는 데 주력해왔다.<sup>[1]</sup> IP 주소 탐색은 인터넷 통신에서 패킷 처리 능력을 결정하는 중요한 요소이다. 인터넷의 속도는 통신 선로가 아무리 빠르더라도 패킷을 목적지 주소에 따라 적절한 출력포트로 라우팅(routing)을 해주는 IP주소 탐색 처리능력이 작으면, 고속의 인터넷 통신이 불가능해진다.

현재의 가변 길이 IP 주소는 CIDR(Classless Inter Domain Routing)을 기본개념으로 하는 라우터로 구성되어 네트워크의 상호연결 장치를 통해 직접 접속이 가능한 경로만을 유지하고 있다. CIDR 주소 할당 방식의 특징은 기존의 클래스(class)기반에서 목적지 IP 주소와 해당 IP 주소의 연결하고자 하는 프로토콜(protocol)의 구분 점을 찾아내기 위해 IP 주소

의 처음 3bit를 사용했던 것과 달리, 기본 구분 점을 없애고 IP 주소에 대해서 주소의 길이를 자유롭게 조정하여 사용한다는 것이다. 이로 인해 IPv4의 주소 공간을 효율적으로 할당할 수 있게 구성되었다. 또한, 네트워크 주소의 CIDR 블록을 지원하여, 하나의 경로지장 엔트리(entry)가 수천 개의 기존의 클래스 기반 경로를 대치 할 수 있다. 즉, 라우터에서는 동일한 출력포트로 도달 될 수 있는 IP 주소들을 하나의 집합으로 묶어서 하나의 프리픽스(prefix)로 표현하여 라우터의 룩업 테이블의 한 요소로 사용함으로써, 탐색 테이블의 크기를 대폭 줄일 수 있었다.<sup>[2][3]</sup> CIDR의 논리적 주소 체계는 LNPM(Longest Network Prefix Matching)이 요구되어, IPv4 주소 체계에서는 32bit 마스크(mask)로 처리된다. 이 마스크의 크기는 IP 주소의 유효한 주소를 결정한다. 즉, 경로설정 프로토콜은 네트워크 프리픽스(network prefix) 정보를 전달해야 한다. 또한, 이와 비슷한 개념의 VLSM(Variable Length Subnet Mask)이 있다. CIDR과 VLSM은 둘 다 IP 주소 공간의 일부를 잘라서 사용한다는 점에서는 근본적으로 동일하나 VLSM은 한 기관에 이미 할당된 주소 공간상에서 재사용이 가능하며, 글로벌 인터넷(global internet)에서는 이 관계가 보이지 않는다는 점이 다르다. CIDR 경로 탐색은 직접적인 테이블 매칭이 아니므로, 큰 테이블에서의 소프트웨어기반의 알고리즘 방법은 매우 복잡하게 된다. 따라서 CIDR 라우팅 룩업 알고리즘(routing lookup algorithm)을 하드웨어로 구현하여야 한다. 즉, 라우팅 룩업 테이블은 고속화 알고리즘을 기반으로 하는 하드웨어방식으로 처리하고, 포워딩 엔진(forward engine)과 제어 장치(control engine)의 분리로, 분산 또는 병렬화된 전송방식을 채택하여야 한다. 현재 기가비트(giga-bit)급 라우터는 주로 백본(back bone) 망에서 전송되는 데이터를 처리하는데 사용되고 있으나, 룩업 테이블의 개수는 45,000개 정도인 것으로 알려져 있다.<sup>[4]</sup> 현재의 인터넷 발전 추세로 볼 때 폭주하는 데이터를 원할하게 처리하기 위해서 추후에는 소형의 에지(edge)급 라우터에 주로 사용될 것으로 추정된다. 특히, IP 주소 탐색은 IPv4 체계의 효율적인 사용을 위하여 채용된 가변 프리픽스 길이(variable prefix length)를 수용해야 하므로, 테

이들 탐색의 최대시간과 평균시간이 동시에 짧아야 하며, 라우팅 테이블의 갱신시간이 짧은 룩업 알고리즘과 구현 방법이 필요하다. 신속한 룩업 테이블의 갱신이 이루어지지 않으면, 부정확한 경로설정이 발생할 수 있으며 룩업 테이블의 갱신이 빠르면, 이러한 문제들이 해결 될 수 있다.

본 논문에서는 ASIC기반의 IP 경로설정 장치에 대해 빠른 주소 룩업 처리를 위한 새로운 룩업 테이블을 설계 검증하였다. 설계한 파이프라인 룩업 테이블은 고속 LPM 탐색을 위한 구조로서 갱신 시간이  $O(1)$ 으로 짧으면서도, LPM 탐색율이 높고, 복잡도도 높지 않은 새로운 방식의 파이프라인 구조로 설계되었다.<sup>[5]</sup> 룩업 테이블은 3단계의 파이프라인 처리를 하고, 또한, IP 주소 키 필드 분할 수를 8개의 블록으로 나누어 병렬탐색을 수행한다. 1단계에서는 분할된 블록별로 탐색을 하고, 2단계에서는 선택된 블록에서 각각의 비트(bit) 단위 탐색을 수행한다.

본 논문의 구성은 다음과 같다. II장에서는 LPM 탐색에 대한 기존 방법을 살펴 본 다음, III장에서는 새로운 탐색 방법인 제안한 파이프라인 탐색구조를 기술하고, IV장에서는 실험에 대한 성능을 정리하면서, 또한 효율적인 파이프라인 탐색구조를 논하고<sup>[6]</sup>, 마지막으로 V장에서 결론을 맺는다.

## II. 기존 LPM 탐색 방법

기존의 고정길이 IP 주소의 탐색에는 패킷의 목적지 IP 주소와 정확하게 일치하는 엔트리를 룩업 테이블에서 찾아내는 EM(Exact Matching) 탐색을 사용하였다.<sup>[7]</sup> EM탐색은 해싱(hashing), 이진 탐색(binary search), 또는 일반적인 CAM 장치를 사용해 빠르게 수행 할 수 있었다. IP 룩업 테이블 속도는 저장장치(memory) 접근 횟수와 속도에 의해 결정된다. 탐색을 위해 요구되는 메모리 접근 횟수는 룩업 알고리즘에 따라 다르며, 룩업 엔진(lookup engine)의 물리적인 동작속도 향상은 소프트웨어와 하드웨어 기반의 방법으로 나눌 수 있다. 전자는 고속의 RISC(Reduced Instruction Set Computer) CPU를 사용하고, 개선된 알고리즘을 통해 IP 탐색을 위한 메모리 접근 횟수를 최소화함으로써,

처리속도를 증가시키는 방법이다. 초기 개발비용이 적고 매칭 알고리즘에 유연성을 줄 수 있다는 장점은 있지만, 고속의 CPU 가격이 비싸다는 단점과 캐시 히트 율(cache hit traffic pattern)에 따라 라우터 성능이 좌우된다는 단점이 있다. 후자는 매칭 알고리즘을 ASIC화하거나 매칭 알고리즘에 대한 하드웨어와 메모리를 원 칩(system on chip)화 함으로써 고속의 탐색을 수행하는 방식으로, 처리 속도가 빠르고, 양산시 포워딩 엔진의 가격이 저렴해진다는 장점이 있다. 하드웨어 기반으로 전형적인 CAM을 사용하는 방법도 있으나 확장성 문제로 인해 백본용(대용량) 라우터에는 적합하지 않다.<sup>[7][8][9]</sup>

전형적인 CAM구조로 설계하는 IP 주소 탐색 방법에 McAuley가 B1, B2, B3, T1, T2 및 T3의 6가지 방법을 제안했다. B1, B2, B3의 방법은 B-CAM(binary CAM)을 사용하는 방법이고, T1, T2, T3는 T-CAM(ternary CAM)을 사용하는 방법이다.<sup>[7][8]</sup> 여기서 B1은 EM 탐색에 적용할 수 있는 방법이고, 그 외의 방법은 LPM 탐색에 적용할 수 있는 방법이다. 여기에서는 기존 탐색 알고리즘의 간략한 성능과 전형적인 CAM의 구조, 그리고 B-CAM과 T-CAM의 기본 동작원리에 대해 좀더 자세히 살펴보도록 하겠다.

### 1. 이진 탐색 알고리즘

기존의 이진 탐색 알고리즘 방법은 메모리의 접근 횟수가 많고, 주소의 프리픽스 비트 수가 증가하고, 라우팅 테이블 내의 엔트리 증가 등, 확장성의 문제가 있어 기가 비트 급 라우터에 적용될 수 없다. 이진 탐색 알고리즘 기반에서 패트리샤 트리(Patricia tree)방법의 경우 최대 32번(IPv4)의 메모리 접근이 필요하게 된다. 메모리 속도는 DRAM(Dynamic Random Access Memory)의 경우 60ns에서 100ns이며 SRAM(Static Random Access Memory)의 경우 10ns에서 20ns가 소요된다. LPM 방식에서 메모리로 DRAM을 사용하는 경우 패킷 당 최대 3.2  $\mu$ s 시간이 소요된다. LC-Trie(Level Compression-Trie)<sup>[10]</sup>를 이용한 탐색 알고리즘은 기본적인 이진 트리(binary trie)의 깊이(level)를 줄임으로써, 트리의 자 노드(leaf node)에 도달하기 위해서 거쳐야 하는 내부 노드의 수를 줄이고자

하는 접근 방법이다. 해싱 기법을 적용한 탐색과 같이 프리픽스 길이마다 별도의 해시 테이블(hash table)을 두고 목적지 IP 주소의 LPM을 빠르게 찾을 수 있도록 이진 트리를 적용한 기법도 있다.<sup>[11]</sup> 이러한 주소 탐색 방법이 다양하게 연구되어 디스크(disk)와 주 저장장치(main memory)사이에서 메모리 계층구조를 고려하여 다중 탐색 트리(multi-way search tree) 알고리즘이 고안되었다. 즉, 저속의 메모리와 상대적으로 고속인 메모리를 고려하여 한 번에 디스크에서 주 저장장치로 읽어 들인 디스크 블록(disk block)을 이용하여 추가적인 저속의 디스크 접근 없이 분기 수를 줄여 주 저장장치의 속도를 늘리는 방법이다. 최근에 테이블의 압축과 약간의 계산을 통해 직접참조 기법에서의 많은 메모리의 요구량을 줄이면서 빠른 참조 기법이 제안되었다. 이는 IP 주소 공간(232)에 해당하는 메모리에 라우팅 테이블을 저장하는 직접 참조기법과 이진 트리를 적절히 조합한 다중 트리(multi-way trie) 기법 등이 있다.<sup>[12]</sup>

2. 전형적인 CAM기반의 탐색

전형적인 CAM은 그림1과 같은 구조를 갖고 있다. CAM은 탐색을 위한 키 필드(key field)와 데이터 필드(data field), SAR(Search Argument Register, 탐색 레지스터), MR(Mask Register, 마스크 레지스터), MDR(Memory Data Register) stage1, MDR stage2 및 기타 주변회로 등으로 구성된다.

탐색 키 필드는 CAM 셀로 구성되어 있으며, 탐색 키 필드는 w bit 길이의 워드 n개로 구성된다. IP 룩업 테이블에 사용될 경우 탐색 키 필드에 룩업 테이블의 각 엔트리의 IP 주소가 저장된다. 데이터 필드는 탐색기의 각 워드와 관련된 데이터를 저장하는 곳으로 탐색 키 필드에서 특정 워드가 탐색되면, 그 워드와 관련된 데이터를 출력하게 된다. IP 주소 룩업 테이블인 경우에는 IP 주소에 따른 출력포트의 데이터를 출력하게 된다. SAR은 찾고자하는 데이터의 패턴(pattern)을 저장하는 레지스터(register)이다. MR은 저장된 w bit 길이의 워드 데이터 중 매치 되어야만 하는 비트를 지정하는 레지스터이다.

SAR에 탑재된 패턴 중 탐색 키 필드의 데이터와 어느 비트가 매치 되어야 하는가를 MR을

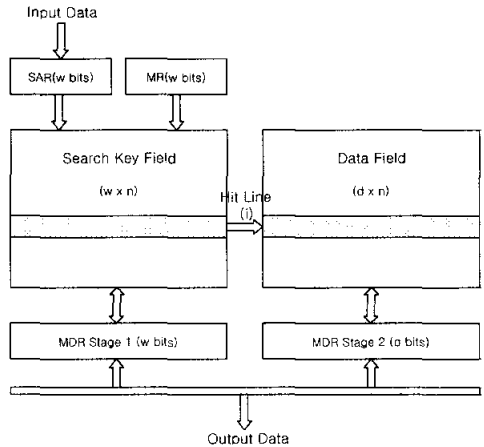


그림1. 전형적인 CAM

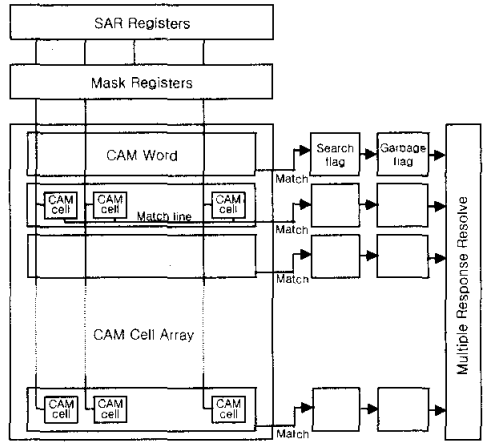


그림2. 기존 CAM 셀 배열

통해 지정할 수가 있다. 즉, MR의 비트 중 1인 비트는 유효한 비트이고, 0인 비트는 유효하지 않는 비트임을 의미한다. 찾고자 하는 패턴이 탐색 키 필드에서 찾아지면, 결과 신호로서 히트신호(hit)가 발생한다. 그림1에서는 빗금 친 부분의 워드가 찾아졌다는 것을 보여 주고 있고, 히트 라인(hit line) i가 데이터 필드의 관련된 데이터를 접근하는 신호로 가해진다. 그러면 해당 데이터가 MDR stage2를 통해 출력되게 된다. MDR을 통해서 탐색 키 필드의 데이터 및 데이터 필드의 데이터를 읽고 쓸 수 있고, 데이터를 추가하고 삭제할 수 있다. 여기서 탐색 key field, SAR, MR 및 MDR stage1의 폭은 w bit이고, data field 및 MDR stage2의 폭은 d bit이다. 전형적인 CAM에서 메모리 셀

배열과 외부 레지스터 사이의 관계가 그림2에 나타나 있다. 배열의 메모리 셀은 1bit 크기의 CAM 셀 이고 입력에 SAR 레지스터와 마스크 레지스터, 그리고 출력에 각 배열에서의 메모리 셀들의 탐색 결과로서 매치 신호가 출력된다. 그리고 다중 매치 결과에 대한 우선순위 선택 회로 등이 있다.

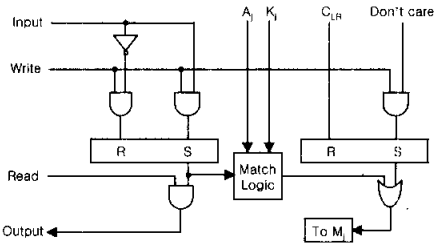


그림3. 전형적인 CAM 셀

### 3. 전형적인 CAM 셀 구조

그림3은 B-CAM 셀 구조에 don't care를 저장할 수 있도록 설계한 전형적인 CAM 셀 회로이다. B-CAM 동작 구조는 플립플롭(flip flop) 기억소자  $F_{ij}$ 와 읽기, 쓰기 및 셀의 매치를 위한 회로로 구성되어 있다. 입력 비트는 쓰기 동작 시 플립플롭 기억소자에 전송되며, 저장된 비트는 읽기 동작 동안 읽혀진다. 매치 논리 회로는 플립플롭 셀의 내용과 마스크 되지 않은 인자(argument)의 비트를 비교하여  $M_j$ 를 세트(set)시키는 동작을 한다. 또 다른 형태의 CAM구조는 전형적인 B-CAM 셀에 don't care bit를 저장할 수 있도록 설계 되어 T-CAM으로 동작하는 구조이다. don't care bit를 저장한다는 것은 CAM 셀 내에서 입력되는 탐색 데이터와는 관계없이 무조건 매치 시키는 것을 의미한다. 즉, 마스크(masking)하는 것을 말한다. 즉, CLR 신호가 '1'이고 don't care 신호가 '0'이면 '0'이 저장되므로 일반적인 동작인 읽기 및 쓰기 그리고 매치동작을 수행한다. CLR신호가 '0'이고 don't care신호가 '1'이면 '1'을 저장하고 입력되는 탐색 데이터와 관계없이 무조건 매치 신호를 '1'로 만들어 매치가 되게 한다. 결국, don't care를 저장하는 것과 동일하게 된다. 이 방식은 데이터의 내용으로 병렬 탐색하기에 알맞도록 되어 있으며, 탐색은 전체 워드 또는 한 워드내의 일부만을

가지고 비교 할 수 있다. 매치 동작은 앞서 설명한 것과 같이 B-CAM동작과 같다.<sup>[13]</sup> 그림4는 전형적인 CAM의 매치 탐색에 대한 동작도이다. 그림4에서 동작을 살펴보면 신호 CLK가 발생되고, 입력신호 DIN이 발생되면 신호 SRT\_EN에 의해 매치 탐색이 시작된다. 입력신호 DIN은 탐색 할 데이터이고, IP 탐색에서는 탐색할 IP 주소에 해당된다. 매치 신호인 MATCH가 high('1')가 되고, 매치된 주소 값 MATCH\_ADDR 신호가 출력된다.

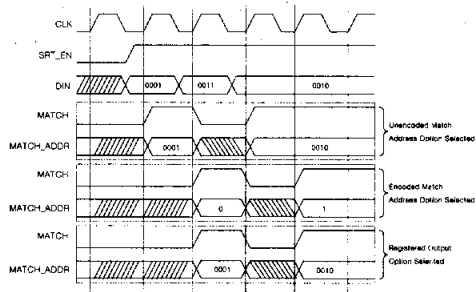


그림4. CAM의 매치동작 탐색

### III. 제안한 파이프라인 CAM 탐색구조

LPM 탐색을 위한 방법으로 CIDR을 사용하면 라우터의 룩업 테이블(lookup table)은 IP address/mask, port)로 표현되는 엔트리들로 구성된다. 그림5에서 보면 마스크(mask) 길이는 프리픽스(prefix)의 크기이고, IP주소/마스크 길이의 결과는 프리픽스 정보를 나타낸다. 만일, 입력된 패킷의 IP주소가 128.32.195.1이라 할 때, 라우팅 테이블에서 매치되는 엔트리가 128.32.1.5/16,port1:1000000000100000\*, 128.32.225.0/18,port3:100000000010000011\*, 128.0.0.0/8,port5:10000000\*이면, 이들 중 왼쪽로부터 가장 길게 매치되는 엔트리는 두 번째의 128.32.225.0/18,port3이며, 이를 LPM이라 부른다. 이 LPM을 라우터의 룩업 테이블에서 찾는 작업을 LPM 탐색이라 한다. IP 패킷을 라우팅 할 때는 룩업 테이블에서 이 LPM을 찾아 그에 해당하는 출력포트인 port3로 입력 패킷을 출력함으로써 라우팅을 수행한다. LPM 탐색의 어려움은 입력 패킷의 IP 주소의 프리픽스 길이를 미리 알 수 없는 것이 문제인데, 더구나 입력 패킷에 대해 여러 개의 매치 가능한 프리

픽스 중 가장 길게 매치되는 프리픽스를 찾아 내야 하기 때문에 탐색시간이 오래 걸려, 고속의 IP 주소 룩업에서 LPM 탐색은 병목현상을 일으키고 있다.

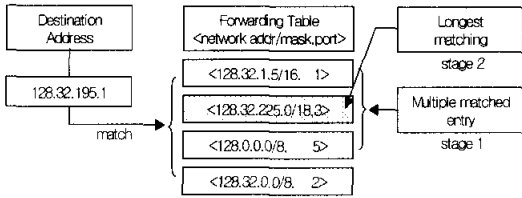


그림5. LPM 탐색의 예

본 논문에서 제안하는 파이프라인 룩업 구조는 기존의 CAM에서 B-CAM의 공간적인 병렬 처리로 인해 복잡도가 매우 높은 단점과, T-CAM의 룩업 테이블에서의 정렬을 유지해야 하는 단점을 제거하기 위해, 전형적인 B-CAM셀을 이용해 크기를 최소화하고, 병렬성 대신에 파이프라인 구조를 적용하여, 복잡도를 줄이고 동시에 룩업 테이블의 정렬을 유지해야 하는 단점도 제거할 수 있는 새로운 방식의 룩업 테이블 구조이다. 설계한 파이프라인 룩업 테이블은 B-CAM의 키 필드와 데이터 필드를 분리해 3단계의 파이프라인 구조로 이루어진다. 파이프라인 처리이기 때문에 하나의 입력 IP 주소에 대한 LPM 탐색 시간은 더 오래 걸리지만, 단위 시간당 처리하는 IP 주소의 수인 LPM 탐색율(throughput)은 더 올릴 수 있다.

### 1. 키 필드 블록 탐색 구조

그림6은 설계된 블록 매치 탐색 구조 도이다. 그림6에서 보면 입력에 룩업 테이블 엔트리의 각 CAM 셀에 저장할 엔트리 데이터 CAM DATA 신호와 탐색을 위한 IP 주소인 DATA TO COMP 신호가 있고, 4bit 크기로 구성된 CAM 셀 블록이 있다. 이 4bit 블록이 키 필드이고, 키 필드 탐색은 COMPARE 논리회로 블록에 의해 탐색되고, DATATOCOMP 신호가 들어오면 각 셀 들은 동시에 병렬로 탐색된다. 매치 결과는 4 입력 논리 곱 게이트로 입력되고 4bit 모두가 매치 될 경우에만 논리 곱 게이트 출력이 high('1')가 된다. STARTCOMP 신호에 의해 MATCH\_BIT 조합 논리회로로 입력된다.

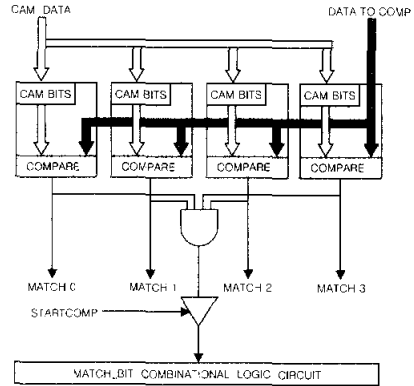


그림6. 키 필드 블록 탐색

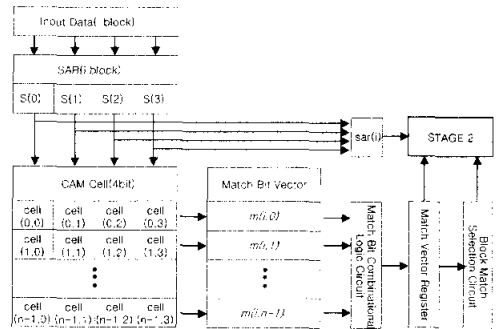


그림7. 단계1에서의 키 필드 블록매치 탐색

### 2. 단계1의 세부 구조

단계 1의 키 필드 블록 매치를 위한 탐색 방법을 그림7에 나타내었다. 탐색해야 할 데이터를 저장하는 레지스터 SAR, 룩업 테이블의 해당 키 필드의 데이터를 저장하는  $4 \times n$  CAM 셀 배열, 각 엔트리 세그먼트(segment) 단위로 탐색해야 할 데이터와의 매치여부를 나타내는 match bit vector로 이루어진다. 또한 연속적인 match 여부를 확인하기 위한 match bit 조합회로와 단계2에서 bit match 탐색을 위한 키 필드의 match block 탐지회로로 구성된다. 외부에서 들어온 패킷은 4bit 크기로 분할되어 각 키 필드 블록의 SAR에 인가된다. 여기서 CAM 셀 배열의 크기는  $4 \times n$ 이며 n은 엔트리 개수이고,  $S(k) = SAR(4*i+k)$ ,  $C(k,j) = CAMcell(4*i+k,j)$ 로 나타낼 수 있다. S(k)의 S는 SAR을 m개로 균등 분할한 하나의 세그먼트를 나타내고, k는 키 필드(key field)이다. S는 4bit로 이루어지며, i 번째 키 필드 블록의 S에는

4i 번째 비트에서 4i+3 번째 비트까지 인가된다. 룩업 테이블의 j 번째 엔트리의 i 번째 세그먼트 데이터는 4 x n CAM 셀 배열에서의 j 번째 행의 데이터가 된다.

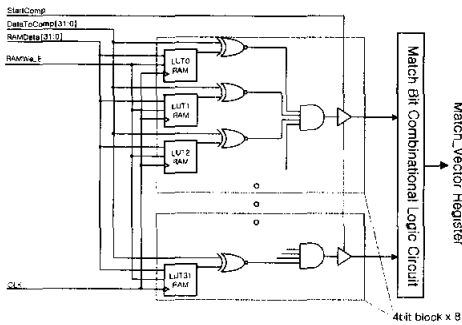


그림 8. 32bit CAM 셀 엔트리 탐색구조

해당 키 필드 블록으로 매치 벡터 정보와 SAR의 데이터를 전달한다. 또한, 매치 비트 조합회로가 n개 있으며, 각 매치 비트 조합회로는 매치 벡터의 각 한 비트씩 담당하여 데이터를 처리하고 있다.

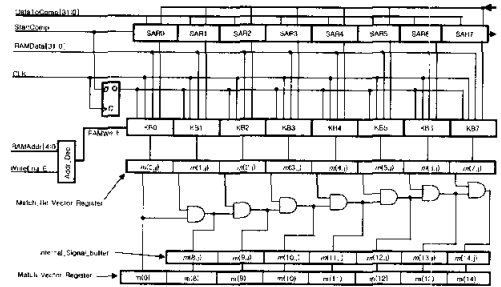


그림 9. 매치 벡터 생성회로

### 2.1 키 엔트리 탐색 구조

그림8은 1bit RAM 셀을 이용한 32bit CAM 블록을 설계한 회로이다. 이 구조는 각 RAM 셀에 탐색회로를 넣어 CAM과 같이 비트 단위 탐색이 가능하도록 설계한 구조이다. 4bit 크기의 키 필드 블록을 구성하여 동시에 병렬 탐색할 수 있도록 설계하였다. IPv4를 기준으로 할 경우 주소의 크기는 32bit이므로 이러한 키 필드 블록이 8개가 존재하며 이를 키 엔트리(key entry)라 한다. 키 엔트리 탐색 방법은 각 키 필드 블록의 매치 정보를 매치 비트(match bit) 조합 논리회로에 의해서 연속적인 매치정보를 구하고, 최종적으로 단계2의 매치 벡터(match vector) 레지스터에 저장하게 된다. 매치 벡터 레지스터는 블록 매치(block match) 정보를 구하기 위한 레지스터이고, 비트 단위 탐색을 위해 단계2에서도 사용된다.

### 2.2 매치 벡터 생성을 위한 상세도

그림9는 단계1의 매치 벡터 생성회로의 상세이다. 매치 비트 벡터 레지스터와 매치 벡터 레지스터는 각각 n개의 신호로 구성되며, 매치 벡터 생성회로는 매치 비트 벡터로부터 j번째 매치 비트들을 입력받아 매치 벡터의 j번째 신호를 합성하는 과정을 보여주고 있다. 이러한 매치 벡터 생성회로는 단계1의 각 키 필드 블록에서 넘어온 매치 비트들을 이용해서 단계2의 비트 매치 탐색을 위한 매치 벡터를 구해

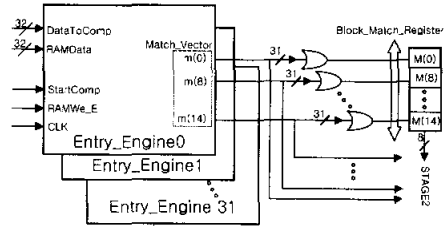


그림 10. 각 엔트리 블록 매치 탐색도

### 2.3 블록 매치 탐색구조

그림10은 단계1의 각 엔트리를 32개로 구성하여 룩업 테이블을 구성하였다. 블록 매치 레지스터 정보를 구하기 위한 구조로 각 엔트리에서 매치 벡터 정보를 얻고, 왼쪽부터 가장 길게 매치된 엔트리 정보를 얻어 블록 매치 레지스터에 전달한다. 블록 매치 레지스터 정보는 단계2를 위한 데이터이다. 각 신호에 대한 정보를 살펴보면 DataToComp 신호는 비교하기 위한 데이터로 탐색할 IP 주소이다. RAMData 신호는 각 엔트리의 룩업 테이블에 라우팅 데이터를 저장할 때 사용되고, RAMWe\_E 신호와 CLK, StartComp는 제어신호이다. 먼저 RAMWe\_E과 CLK 신호가 '1'일 때 RAMData 신호를 통해 엔트리 데이터가 써지고, DataToComp 신호에 탐색할 IP 주소가 입력된 후, StartComp신호가 '1'일 때 탐색이 시작된다.

### 2.4 블록 선택신호 생성회로

그림11의 블록 선택신호 발생회로는 매치 벡

터 신호를 wired-OR해서 생성된 M(0)에서 M(14)의 블록 매치 레지스터에 의해서 발생된다. 이는 단계2 에서 비트 단위로 탐색 할 블록을 결정하는 신호로서 e0,e1,...,e7은 각각 키 필드 블록0, 키 필드 블록1, ..., 키 필드 블록7을 선택하는 신호이다. 블록 선택신호의 어느 비트가 '1'이면, 해당 블록이 선택되고, '0'이면 선택되지 않음을 의미한다. 또한, 블록 선택 신호는 단지 한 비트만 '1'이 되고, 나머지는 모두 '0'이 된다.

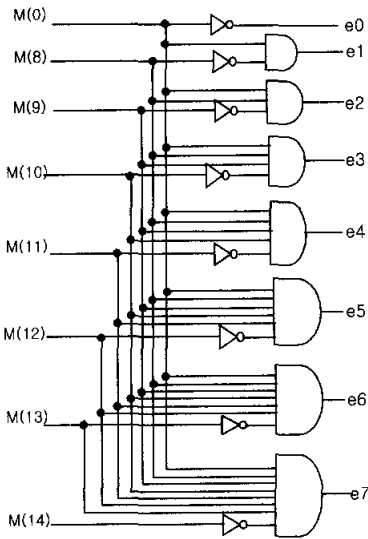


그림 11. 블록 선택신호 생성회로

### 3. 단계2의 세부 구조

그림12는 그림10의 단계1에서 단계2를 연결하여 비트 매치 탐색구조가 포함된 전체 룩업 테이블 구조이다. 단계1에서 블록 매치 정보를 얻고 단계2에서 비트 매치 정보를 얻기 위한 구조로 비트 탐색을 위한 키 필드 블록을 선택하여 비트 매치 탐색 정보를 얻어, 단계3에서 최종적으로 매치된 IP 주소와 해당 포트번호를 출력하게 된다. 히트벡터(hit vector) 발생회로는 포트번호를 선택하기 위한 조합 논리 회로이고, 또한 단계2의 룩업 테이블에서의 엔트리 데이터는 단계1에서의 엔트리 데이터가 쓰여 질 때 동시에 미리 가져온다. 이는 단계1과 단계2의 처리속도가 달라 파이프라인 탐색을 하기 위한 조치이다.

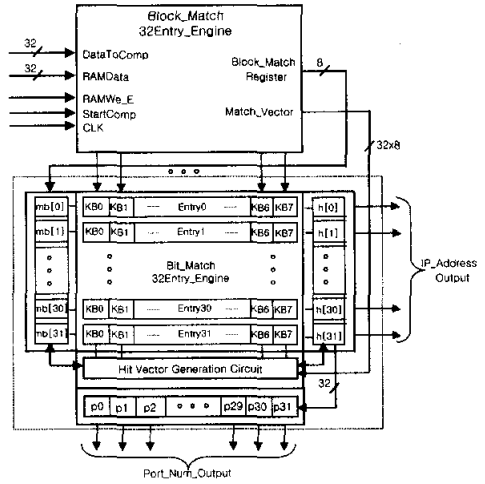


그림 12. 설계된 룩업 테이블의 구조

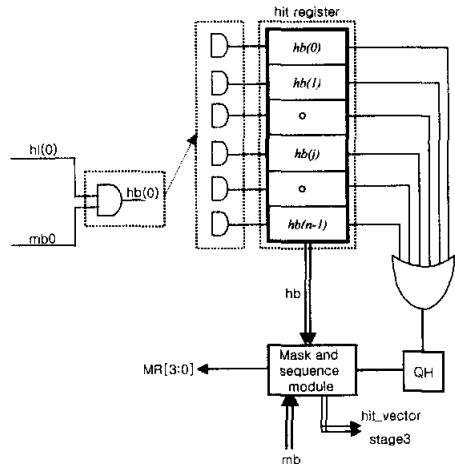


그림 13. 히트 벡터 생성회로

#### 3.1 히트 벡터 생성회로

그림13은 단계3을 위한 히트 벡터 생성회로를 나타내었다. 히트 벡터 생성회로는 독립적으로 서로 다른 패킷의 세부 비트 탐색을 수행하며, 동시에 동작 할 수 있다. 생성회로는 매치 비트 벡터(mb)와 히트 라인 벡터(hl), 히트 레지스터(hit register), QH신호 생성회로, 마스크 및 시퀀스모듈(mask and sequence module), 마스크 레지스터(MR)로 구성된다. 각 엔트리에서 탐색된 결과는 hl(0),hl(1), ... ,hl(j), ... ,hl(n-1)로 이루어지는 히트 라인 신호이며, 이 신호들은 매치 벡터 레지스터의 신호들과 각각 논리곱(AND)이 되어 히트 레지스터에 저장된다. 히



트 레지스터의 모든 비트들은 논리 합(OR)을 수행하여 QH신호를 만든다. QH신호가 '1'이면 현재 비트까지 매칭되고 있는 엔트리가 있음을 의미하고, '0'이면 없음을 의미한다. 마스크 및 시퀀스 모듈은 QH신호에 따라 마스크 레지스터의 매스킹(masking)을 조절해 가면서 비트 단위 탐색을 진행시키는 역할을 하며, 단계3으로 보내는 히트 벡터 정보를 생성한다.

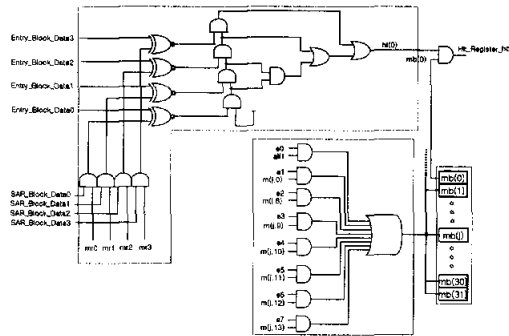


그림 14. 마스크 및 시퀀스 회로 모듈

### 3.2 마스크 및 시퀀스 회로

그림14는 마스크 및 시퀀스 모듈의 제어흐름에 따라 동작하는 조합 논리회로의 세부회로이다. 단계2는 4bit의 키 필드 블록을 이진 탐색 알고리즘을 이용한 탐색과정으로 2번의 탐색이 이루어진다. 첫 번째 탐색 사이클은 탐색 키(SAR block) 레지스터에 단계1로부터 입력된 SAR데이터가 탑재(load)되고, 매치 벡터 레지스터(mb)에는 매치 비트 벡터 정보가 탑재되며, 마스크 레지스터(mr)에는 초기 mr0에서 mr3으로 '1100'이 차례로 탑재된다. 마스크 레지스터에 '1100'이 탑재되는 것은 4bit 중에 가운데부터 우선 매치 여부를 판별하기 위해서이다. 이러한 준비과정을 마치고 키 필드 블록을 탐색한다. 각 엔트리의 히트 출력인 히트 라인 신호 hl을 매치 벡터 레지스터와 논리 곱 연산을 수행하여 히트 레지스터에 저장한다. 이전 단계의 매치 결과와 현 단계의 매치 결과가 동시에 '1'이 되어야만, 현재 단계까지의 누적 매치 결과가 '1'이 된다. 즉, 그림13에서 QH신호가 '1'이면 4bit중 가운데에서 오른쪽으로 탐색해야 하고, 0이면 왼쪽으로 탐색을 해야 한다. 이때 다시 QH신호가 '1'이면, 마스크 레지스터에 '1100'을, QH신호가 '0'이면, 마스크 레지스터에 '1000'을 탑재하여 다음 탐색 사이클을 진

행해야 한다.

## IV. 실험 결과 및 분석

### 1. 레지스터 파일 구조

그림15는 룩업 테이블 갱신을 위한 레지스터 파일(register file) 구조이다. 최초 엔트리에 데이터를 입력하는 과정과 탐색이 끝난 후 엔트리 갱신을 위한 구조로, SAR을 위한 데이터를 미리 저장해 놓은 다음, 제어신호에 의해 SAR에 갱신하는 데이터를 입력한다.

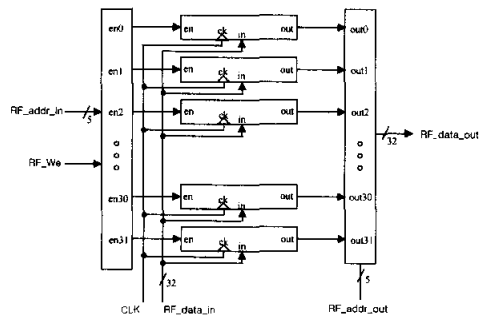


그림 15. 레지스터 파일 배열 블록도

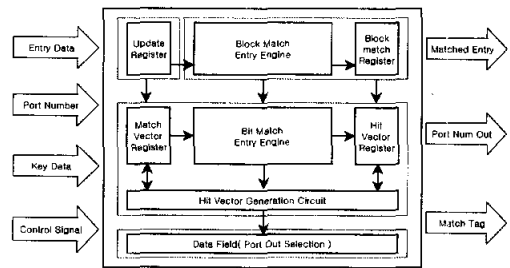


그림 16. 설계된 룩업 테이블 블록도

### 2. 전체 룩업 테이블 블록도

그림16은 레지스터 파일구조가 포함된 전체 블록도이다. 설계 사양은 입력으로 엔트리 정보(Entry Data), 연결 포트 번호(Port Number), 탐색할 주소 정보(Key Data), 제어 신호(Control Signal) 등이 있고, 출력에는 매칭된 IP 주소(Matched Entry), 출력포트 번호(Port Num Out), 그리고, 갱신을 위한 매치 태그 정보(Match Tag)가 출력된다. Matched Entry는 룩업 테이블에서 매칭된 목적지 라우팅 주소이고, Port Num Out은 해당 프로토콜

포트 번호이다. 내부 블록은 갱신을 위한 Update Register, Block Match Entry Engine, 그리고 각각의 엔트리별로 블록 매치 정보를 저장하는 Block Match Register과 Bit Match Entry Engine, 그리고 비트 탐색은 Match Vector Register와 Hit Vector Register에 의해 탐색된다. 비트 탐색이 끝나면, 히트 벡터(Hit Vector) 발생회로에 의해서 최종적으로 매치된 목적지 IP 주소를 단계3인 데이터 필드(Data Field)에서 선택해 출력하고, 해당 포트번호(Next Hop)가 출력된다. 매치 태그 정보는 하나의 IP 주소 탐색이 끝나고 갱신을 위한 엔코더(encoder) 정보이다.

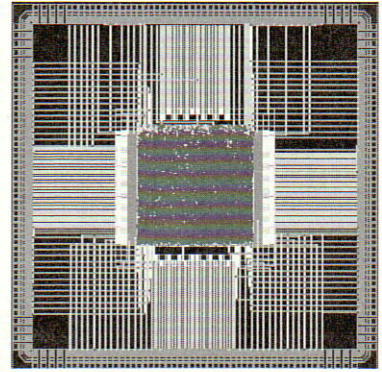


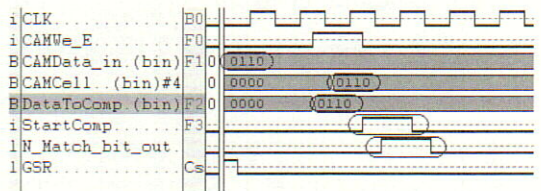
그림 17. Cadence에서의 physical layout 도면

3. 합성 및 레이아웃 결과

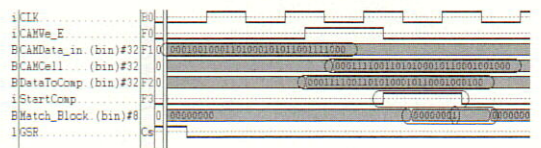
설계가 끝난 후 동작 검증은 Verilog-HDL로 기술하여 Modelsim Simulator로 기능 검증하였고, Synopsys 합성(synthesis)용 라이브러리를 사용하여 논리회로 합성을 하였다. 표.1은 ASIC구현에서의 실험특성 이다. 32엔트리 룩업에서의 실험특성이며, 클럭 주파수 333Mhz에서 동작하며, 사용된 라이브러리는 1Poly-4Metal의 CMOS 0.35 $\mu$ m/3.3V 공정이며, 입력에 89pin, 출력에 53pin이며, 전체 142개의 I/O와 제어신호 pin을 갖는다. 게이트 수는 표준 게이트 수로 산출하여 51,948개이고, 트랜지스터 수는 207,792개이다. 그림17은 합성 후 추출한 Verilog-HDL netlist 정보를 가지고 Apollo툴에서 배치 및 배선에 대해 Auto P&R 하였고, GDSII형식의 파일정보에 의해 Cadence Virtuoso Layout 툴에서 Physical Layout Cell Merge 단계를 거친 도면이다.

표1. 룩업 테이블 실험 특성

Technology	0.35um / 3.3V (1Poly-4Metal Process SEC)
Function	IP Forwarding
Data Format	In: 89bit Out: 53bit
Total Port	I/O Port: 142bit
Gate Count	51,948
Transistor Count	207,792
Clock rate	333MHz



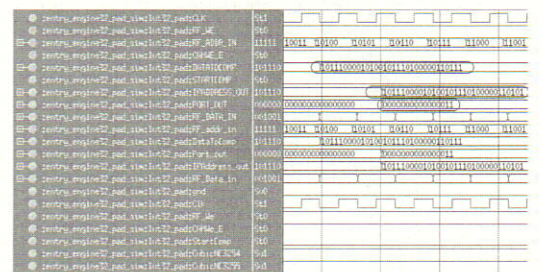
(a) 키 필드 매치 탐색 파형



(b) 블록 선택 탐색 파형

4. 시뮬레이션 결과

그림18은 각 단계 별로 동작 파형과 최종 탐색 파형에 대해서 Postlayout Simulation을 Modelsim 툴에서 수행한 파형이다. (a)는 레지스터 파일구조 블록에서 데이터를 읽어와 각 4bit 크기의 키 필드 블록에서의 탐색과정을 보여준다. CAMData\_in 신호에 '0110'이 입력되고, CAMWe\_E 신호에 의해 CAMCell에 쓰는 과정과 탐색을 위한 데이터 값인 DataTo Comp 신호에 '0110'이 입력되면, StartComp 신호에 의해 탐색이 시작된다. 그 결과 Match\_bit\_out 신호에 매치가 되었다는 신호로 '1'로 된다. (b)



(c) 최종 매치 탐색 파형

그림18. Modelsim에서의 Postlayout Simulation 검증

는 블록 선택 탐지회로에 대한 과정을 보여주고 있다. CAMCell의 가장 오른쪽 4bit('1000')와 DataToComp 신호의 가장 오른쪽 4bit('0100')의 데이터가 달라서, 이 키 필드 블록이 비트 탐색을 해야 할 블록이다. 그리고 (c)는 룩업 테이블 엔트리에 미리 목적지 라우팅 데이터가 저장되어 있고, 입력 정보인 DATATO COMP 신호에 탐색하기 위한 IP 주소"10111000,01010010,11101000,00110111"가 입력된다. STARTCOMP 신호가 '1'로, 즉 탐색 시작이 되면 IPADDRESS\_OUT 신호에 탐색된 주소가 출력되고, 해당 포트번호가 PORT\_OUT 신호에 의해 출력된다. IPADDRESS\_OUT 신호의 출력된 데이터는 "10111000,01010010,11101000,00110101"로 오른쪽에서 3번째 비트까지 매치됨을 알 수 있다. 해당 포트번호는 "00000000000011"로 3이 된다.

5. 실험결과 분석

설계된 파이프라인 CAM의 단계1에서는 한 파이프라인 클럭 주기(clock period), 즉 한 사이클(cycle) 동안에 한 개의 입력 IP 주소만 처리하며, 한번의 CAM 배열 접근이 발생한다. 그러나 전형적인 CAM 배열 접근 시간의 1/2사이클에 해당된다. 단계2에서는 단계1에서 선택한 비트 매치 탐색 블록에 대해 정확한 매칭점을 탐지한다. 매칭점을 탐지하는 방법으로 기존의 이진 탐색을 사용하면 키 필드 블록의 폭이  $b$ 비트 일 때 매칭점을 탐지하기 위해서는  $\log 2b$ 의 회수만큼 CAM 배열 접근이 일어난다. 따라서 키 필드 블록이 하나의 입력 IP 주소에 대해 처리하는데  $\log 2b$ 의 사이클이 소요된다. 그러나, 키 필드 블록들이  $m$ 개가 있을 때, 연속해서 같은 키 필드 블록이 지정되지 않는 한, 병렬로 동시에 동작될 수 있다. 최대  $m$ 개의 키 필드 블록이 동시에 매칭점 탐색을 수행할 수 있다. 따라서 한 사이클 당 처리되는 최대 입력 IP 주소의 수는  $m/(\log 2b)$ 로 표현할 수 있다. 그리고 단계1과 단계2의 룩업 테이블내의 엔트리들이 정렬될 필요가 없다. 룩업 테이블을 갱신 할 때는 각 단계에 있는 룩업 테이블을 동시에 갱신하므로, 한 사이클에 룩업 테이블의 갱신을 완료한다. 따라서 갱신 시간은  $O(1)$ 사이클이다. 다음으로 탐색 시간을 살펴보면 설계된 룩업 테이블, 즉 파이프라인 CAM은 별도의

RAM을 사용하지 않고, 대신 특별하게 설계된 전용 ASIC에서의 포워딩 테이블(forwarding table)로 구현하여, 단지 룩업 테이블에 접근하는 논리회로 지연시간과 탐색에 필요한 시간만 소요된다. 표준 셀  $0.35\mu m$ 의 1Poly-2Metal/3.3V CMOS공정을 사용할 때, 게이트 지연시간(gate delay)은  $0.26ns$ 이다. 따라서, 논리회로 지연시간은  $4.68ns$ 가 소요되고, 탐색 시간은  $3ns$ 로 전체 지연시간은  $7.68ns$ 이다. 이는  $7.68ns$ 의 룩업 테이블 탐색시간이 소요되고,  $130Mpps$ 의 처리 속도으로써,  $130Gbps$ 의 신호처리 능력을 의미하는 것이다. 패킷의 평균 길이가  $1000bits$ 라고 가정할 때, 라우터가  $130Gbps$ 의 전송속도를 내기 위해서는 초당  $130 \times 106$ 번의 탐색이 가능한 것이다.

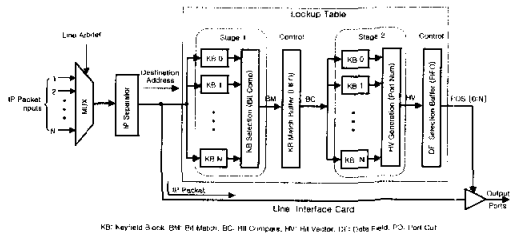


그림19. 효율적인 파이프라인 CAM 구조

6. 효율적인 파이프라인 룩업구조의 성능

본 실험에서는 룩업 테이블 크기가 작게 설계되었다. 그러나 테이블을 크게 하더라도 추가적으로 필요한 논리회로 설계는 상대적으로 적어서 그에 따른 논리회로 지연시간도 거의 늘지 않고, 엔트리를 늘리더라도 설계가 복잡하지 않으며, 속도도 크게 줄어들지 않는다. 단지, 룩업 테이블의 크기를 늘리는 데 있어 높은 집적 설계기술이 수반되어, 대용량 룩업 테이블을 원 칩화하여 집적시키는 것이 관건이다. 그러나 최근  $90nm$  이하의 공정기술이 개발되어 크게 문제가 되지 않을 것으로 본다. 그리고 설계된 파이프라인 CAM구조의 성능에 영향을 미칠 수 있는 요소들이 있다. 먼저, 연속적인 탐색에서 파이프라인 처리과정은 각 단계의 처리 시간이 다르고, 실제 라우터의 패킷 IP 주소의 프리픽스 길이 분포가 균일하지 않다. IP 주소의 프리픽스 길이가 균일하지 않으면, 단계2의 각 키 필드 블록에 걸리는 부하가 균일하지 않아, 파이프라인 처리의 효율을 떨어트릴 수 있다. 이 경우 원활한 동

작을 위해 각 단계에 FIFO형태의 버퍼(buffer)가 있다. 그림19에서처럼 단계1과 2사이에 FIFO구조의 매치 버퍼(match buffer)를 두고, 단계2와 3사이에는 데이터 필드 선택 버퍼를 두어, 파이프라인의 흐름을 깨지 않고 처리되도록 해준다.

표2. 기존 방법과 제안한 CAM의 비교

		existing methods using CAM				Proposed pipelined CAM
		McAuley		Kobayashi		
		B2	B3	T1	VLMP (T1)	
LPM searching rate		$(1/32)P - P$	less than P	P	P/2	higher than P
Update time(cycle)		O(1)	O(1)	O(n)	O(1)	O(1)
Complexity	No. of CAM array Cell	32n	422n	64n	64n	64n
	No. of control modules	1	32	1	2	16

7. 성능 평가

LPM 탐색이 가능한 전형적인 CAM에 대해서 McAuley의 T-CAM의 T2는 B-CAM의 B2보다 2배의 셀을 필요로 하면서도 탐색처리 능력에서는 B2와 비슷하고, T3는 룩업 테이블의 갱신속도를 개선한 방법이나 B3보다 2배의 셀을 필요로 하는 복잡도가 매우 높은 방법이다. 제안한 파이프라인 CAM에 대해서 성능이 좋은 다른 T-CAM과 B-CAM의 비교를 표2.에 나타내었다. 여기서 LPM 탐색시간을 T라 할 때 탐색을 P는 1/TCAM이고, TCAM은 일반적인 CAM의 접근시간이다. n은 2<sup>16</sup>로서, 룩업 테이블의 엔트리 개수이다. 표2에서 B2는 갱신시간은 짧으나, LPM 탐색율이 너무 낮다. B3는 갱신시간은 짧고, LPM 탐색율은 높으나, 복잡도가 너무 크다. T1은 LPM 탐색율은 높고, 복잡도도 크지 않으나, 갱신 시간이 너무 길다. 이에 대해 제안한 파이프라인 CAM은 B3와 T1보다 LPM 탐색율이 크고, 갱신시간은 T1보다 매우 짧다. CAM 셀 수에서는 B3 및 T1보다 작거나 대등하고, 제어회로의 개수에서는 T1보다는 크나 B3보다는 작다. 특히, 최근의 발표 자료인 Kobayashi의 T-CAM(VLMP method)을 보면, 제안한 CAM과 비교할 때 제어회로 수는 1/8이지만, 탐색율에서는 1/2로 일반적인 룩업 엔진(lookup engine)의 경우 적은 지연시간(latency)이나 낮은 복잡도의 차이보다 탐색율이 더 중요시되므로, 전체적인 성능이 떨어진다. 지금까지 분석한 내용을 정리하면, 제안한 파이프라인 CAM은 기존 방법보다 룩업 테이블의 갱신이 매우 빠르며, LPM 탐색율도 높으며, 복잡도도 작다.<sup>[5]</sup>

V. 결론

본 논문에서는 고속 인터넷 라우터를 위한 IP 주소 탐색 기법인 파이프라인 룩업 테이블 구조를 제안 설계하였다. 본 룩업 테이블 구조는 특별한 방법의 하드웨어로 설계하고, 탐색 알고리즘 또한 하드웨어기반으로 설계하여 하나의 ASIC으로 구현함으로써, 메모리 참조시간을 줄이고, 게이트 회로 지연시간을 줄임으로써, 탐색시간을 최적화하였다. 특히 탐색시간과 갱신시간 뿐만 아니라, LPM 탐색의 성공율이 100%이다. 그리고 IPv4인 경우 키 필드 분할 수를 8로 하고, IPv6인 경우 키 필드 블록의 개수를 16으로 하는 것이 최적구조이다.<sup>[6]</sup> 또한 부하가 많이 걸리는 키 필드 블록을 추가적으로 중복 설치하는 것이 최적의 구조라 할 수 있다. 또한, 본 구조는 IPv4 보다는 IPv6에 적용할 때 더 효과적인 성능을 발휘할 것이다. IPv6로 구현할 경우, 단계1에서 1사이클이 걸리고, 단계2에서도 단지 3사이클로 기존의 알고리즘을 이용한 탐색과 비교하여 매우 빠르다. 추후 더욱 빠른 ASIC과 고집적화 된 설계로 테라 비트(tera-bit)급 스위치(switch), 라우터 시스템 구현에 기초를 마련 할 것으로 기대된다.

참고 문헌

[1] Daxiao Yu, Brandon C. Smith, and Belle Wei, "Forwarding Engine For Fast Routing Lookups and updates", *IEEE Global Telecommunications Conference, proceeding on GLOBECOM '99*, vol. 2, pp.1556-1564, 1999.

[2] Y. Rekhter and T. Li."An Architecture for IP Address Allocation with CIDR." *RFC1518*, Sept. 1993.

[3] Marcel W. Waldvogel, George Varghese, JonTurner, Bernhard Platner, "Scalable High Speed IP Routing Lookups", *Proc. of ACM SIGCOM'97*, France, pp. 25-36.


[4] Scott Bradner, "Next Generation routers Overview", *proceeding of Network Interop 97*, 1997

[5] 안희일, 조태원, "고속 LPM 탐색을 위한 파이프라인 CAM 구조(PICAM)", *한국통신학회 논문지*, 제26권 제4호, pp.650-661, 2001

[6] 안희일, 조태원, "PICAM에서의 최적 파이프라

- 인 구조”, *한국통신학회 논문지*, 제 26권 제6호, pp. 1107-1116, 2001
- [7] Anthony J. McAuley and Paul Francis, "Fast Routing Table Lookup Using CAMs", *IEEE INFOCOM'93*, vol.3, pp 1392-1392, March, 1993
  - [8] Anthony J. McAuley, Paul F. Tsuchiya, and Daniel V. Wilson. "Fast multi-level hierarchical routing table using content addressable memory", U. S. Patent serial number 034444. *Assignee Bell Communications research Inc Livingston NJ*, January 1995.
  - [9] M. Kobayashi, T. Murase, A. Kuriyama, "A Longest Prefix Match Search Engine for Multi-gigabit IP Processing", *Proceeding of IEEE International Conference on Communication*, Vol. 3, pp. 1360-1364, 2000
  - [10] S. Nilsson, G. Karlsson, "IP-Address Lookup Using LC-Tries", *IEEE JSAC*, 17(6), pp. 1083- 1092, Jun. 1999
  - [11] Andrei Broder, "Using multiple Hash Functions to Improve IP Lookups". *IEEE INFOCOM*. pp. 1454-1463. 2001.
  - [12] Butler Lampson, Venkatachary Srinivasan, and George Varghese, "IP Lookups Using Multi-way and Multicolumn Search", *IEEE Transaction on Networking*, Vol. 7, No. 3, pp.324-334 June 1999
  - [13] M. Morris Mano, "Computer System Architecture " *Prentice-Hall*, pp. 387, 1994
  - [14] D. Shah and P. Gupta, "Fast Updating Algorithms for TCAMs", *IEEE Micro*, pp. 36-47, Jan.


서 해 준(Hae-Jun Seo) 준회원



2001년 2월 : 청주대학교 전자, 정보통신,반도체공학부 졸업  
2003년 7월 : 충북대학교 전자공학과 석사과정

<주관심분야> 인터넷통신, VLSI 설계, 마이크로프로세서, 영상처리, Embedded system.


안 희 일(Hee-II Ahn) 정회원



1973년 2월 : 서울대학교 전자공학과 졸업  
1996년 2월 : 충북대학교 전자공학과 공학석사  
2001년 8월 : 충북대학교 전자공학과 공학박사  
1976년-1978년: 한국과학기술연구소(KIST) 연구원  
1978년-1999년: 한국전자통신연구원(ETRI) 책임연구원  
1999년 9월-2001년 5월: 충북대학교 전기전자 및 컴퓨터공학부 객원교수  
2001년 5월-현재: 한국조폐공사 기술연구소 정보기술연구팀 수석연구원

<주관심분야> 컴퓨터 구조, IP router, Genetic Algorithm, Smart Card, RFID, Biometrics.

조 태 원(Tae-Won Cho) 정회원



1973년 2월 : 서울대학교 전자공학과 졸업  
1986년 5월 : 미국 루이빌대 전자공학과 공학석사  
1992년 5월 : 미국 켄터키주립대 전자공학과 공학박사  
1973년 8월-1983년 10월: 금성전선(주)  
1977년 1월-1977년 3월: 영국 및 프랑스의 ITT 계열사 연수  
1992년 9월-현재: 충북대학교 전기전자 및 컴퓨터공학부 교수

<주관심분야> 집적회로 설계, 컴퓨터 구조, 저전력 회로, DSP 코어.