High Speed Routing Lookup IC Design for IPv6

Yuan-Sun Chu, Hui-Kai Su*, Po-Feng Lin, and Ming-Jen Chen
Computer Network Laboratory, Department of Electrical Engineering,
National Chung Cheng University, Chia-Yi 621, Taiwan, R.O.C.
Email: *pat@ee.ccu.edu.tw

***Abstract*— With the growth of Internet users and services, the IP address has been exhausted. In order to solve this problem, the short term solution was presented, i.e., CIDR (Classless Inter-Domain Routing). The long term solution for insufficient IP addresses is the IPv6 protocol which was defined by IETF (Internet Engineering Task Force). The length of IPv6 address is 128 bits that can avoid the IP address exhaustion. In this paper, a complete hardware for IPv6 routing lookup architecture is proposed. It is composed of routing lookup ASIC and memory set. In our system, the simple hash hardware is used to reduce the lookup time, and the CAM (Content Addressable Memory) is used to solve the collision problem effectively. In our performance analysis, 91.89% routing entries of the routing table can be searched in one memory access, and the worst case about 10% needs two memory accesses. The CAM in the ASIC is used as cache memory with FIFO replacement algorithm. The routing lookup system approaches 213.4Mlps (109.26Gb/s). It is enough to satisfy the high speed link OC-768 (40Gb/s) with 150000 routing entries.**

I. Introduction

With the growth of Internet users and services, the IP address has been exhausted. In order to solve this problem, the short term solution was presented in 1993, i.e., Classless Inter-Domain Routing (CIDR) [1]. With CIDR, each routing entry is identified by a <prefix, prefix length> pair leading IP address to be more flexible. In this way, the network administrator can flexibly draw up sub-network scale to fit their realistic requirements, and IP address space is assigned more efficiently. The long term solution for insufficient IP addresses is the IPv6 protocol which was defined by IETF (Internet Engineering Task Force) [2]. The IPv6 address length is 128 bits that can avoid the IP address exhaustion.

In recent years, there are many researches for routing lookup. [3], [4] create routing lookup table with trie. Most of them can achieve high average search throughput for IPv4, but they are slow in the updating speed. [5] proposes hierarchical hardware architecture for IPv4 routing lookup, but they cannot suit to 128 bits IPv6 address. [6]–[8] propose routing lookup with CAM, and all match action only needs one clock cycle. But it needs special mechanisms to solve the sorting problem and is more expensive, especially TCAM.

This paper proposes a routing lookup system which contains routing lookup ASIC and off-chip routing table for IPv6. The off-chip routing table is a hierarchical memory architecture with 2 levels, and the scheme is design according to the prefix length distribution of 6Net router's routing table. The first level of off-chip routing table covers 91.89% routing table entries. The ASIC is composed of function unit and a CAM. The CAM is used as cache memory with 1024 entries and guarantees 80% hit ratio by FIFO replacement strategy. In the proposed routing lookup system, routing lookup with 281.69Mlps speed can satisfy the lookup requirement of OC-768. Finally, the routing table only needs 20.04KB TCAM, 10.24KB BCAM, and 29.29MB RAM for 150000 routing entries.

TABLE I
Prefix length distribution of 6Net router

prefix length	entry counts	entry count(%)
19-31	7	1.14
32	444	72.08
33-47	43	6.97
48	81	13.15
64	41	6.66
total	616	100

II. System Architecture

The set of 1-64 bits in the IPv6 global unicast address format is Network ID, and the set of 65-128 bits is Interface ID. Our proposed routing system focuses on the Network ID. Table I shows the prefix length distribution of 6Net router's routing table [9]. In this table, the 91.89% routing entries have prefix length equal to 32, 48 and 64.

The proposed routing lookup system includes a routing lookup ASIC and memory set shown in Figure 1. The ASIC is composed of function unit, cache controller, and cache memory. The memory set stores routing table.

The off-chip routing table is a memory architecture with two-level hierarchy. First level is composed of 3 hash tables, and second level is a pure SRAM table. The 3 hash tables store routing entries with prefix equal to 32, 48, and 64 respectively. In the first level, the routing table contains 91.89% routing entries according to the prefix length distribution of 6Net's routing table.

The routing lookup ASIC has complete functions: inserting, search, update and delete. Moreover, it focus on 1-64 bits of IPv6 address (network ID). 80% hit ratio is guaranteed on the chip, and the whole system can satisfy OC-768 (40Gb/s). The CAM's search operation is parallel and only needs one clock cycle. The SRAM's worst search speed depends on the ram size. Additionally, the CAM is used as the on-chip cache memory for fast search speed. The proposed cache replacement algorithm is FIFO, and it has simplest architecture and nice performance for routing lookup.

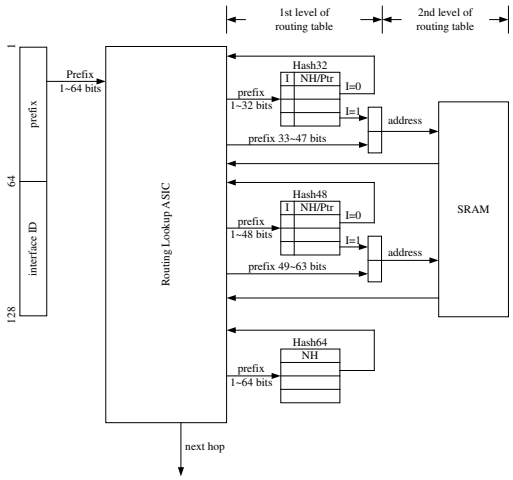

Fig. 1. Routing lookup system.

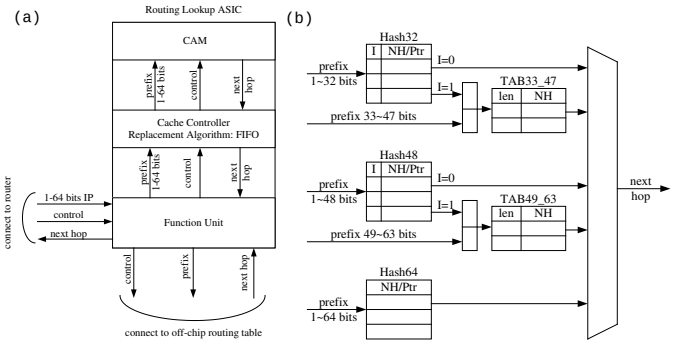

Fig. 2. (a) routing lookup ASIC architecture, and (b) off-chip routing table architecture.

A. Routing Lookup ASIC

Routing lookup ASIC is shown in Figure 2 (a). It is composed of function unit, cache control, and CAM.

Function unit receives the control signal and data from router. The function unit deals with the follow functions: search, insert, update and delete. Function unit controls off-chip routing table to perform the above functions. Besides, it also asks the cache controller to search cache data or update cache information.

The first level of the proposed routing table architecture contains 91.89% routing entries according to the 6'Net router's routing table. Because there is no information about the relationship between the traffic's prefix length distribution and routing table, the performance of routing lookup cannot be guaranteed. Therefore, the cache is designed to guarantee the probability of hitting lookup result in the first search.

B. Routing Table Architecture

Figure 2 (b) shows the proposed routing table which is a memory architecture with two-level hierarchy. First level is composed of 3 hash tables, and second level contains 2 pure SRAM tables.

In order to increase the probability of hitting lookup result in one memory access, the hash tables (Hash32, Hash48, and Hash64) with 32, 48, and 64 prefix lengths are built. Consequently, the first level of routing table contains about 91.89% entries, and the most routing lookup searching actions will be finished in the first search.

The second level of routing table contains 2 tables. TAB33_47 stores the entries with prefix lengths between 33 and 47, and TAB49_63 stores the entries with prefix lengths between 49 and 63. Each routing entry inserted into the second level of routing table also has to store its location index of the routing entry in the previous hash table.

TABLE II

THE ENTROPY OF VARIOUS HASH FUNCTIONS.

Hash Index	Hash Function			
	Bit Extraction	Fletch Checksum	XOR Folding	CRC
12	11.9694	11.9714	11.9753	11.977

III. HASH SCHEME

A. Hash Function Design

[10] defines the entropy as average reduced searching times and uses the entropy to compare various hash functions. A trace of destination IP addresses is done on TANet [11] backbone router. The trace consists of 7.66157 million entries in a period of one hour, and there are 43867 distinct destination addresses. The hash functions used to simulate are Bit Extraction, Fletch Checksum, XOR Folding and CRC. The entropies of above hash functions are computed, and the results are showed in Table II.

In Table II, we can observe that CRC is the best hashing function. However, CRC requires complex computation, and it is more complex than exclusive-OR folding function in hardware. The Fletcher checksum and the bit extraction cannot provide good performance if the patterns of extracted bits are randomly distributed. The XOR folding does not need complex computation and can provide good performance. Because the XOR folding is an excellent hash function and simple to be implemented in hardware, it is used in our scheme.

B. Hash Table Size

Hash function has the problem of hash collision, and the hash collision would cause hash bucket overflow. When the hash table size is bigger enough, the probability of hash collision can be reduced. In this section, the hash table size will be decided by selecting sensible hash table overflow. In [12], an exact probability model for finite hash tables is proposed, and it is used to compute hash table size.

Figure 3 is the hash table chart, k is the total number of IP address, b is number of buckets in hash table, and s is the size of bucket.

In Figure 3, the hash table size is b multiplied by s, so there are two methods to increase the hash table size. Method 1 is increasing bucket numbers (b), and method 2 is increasing bucket size (s). $ExpOverflow(k, s, b)$ is used to compare these two methods' performance, (k, s, b) is assumed

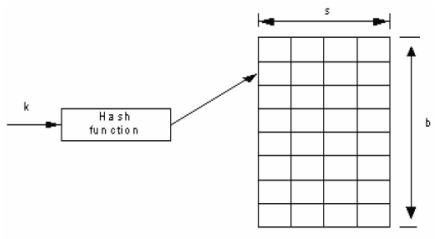

Fig. 3. Hash table chart.

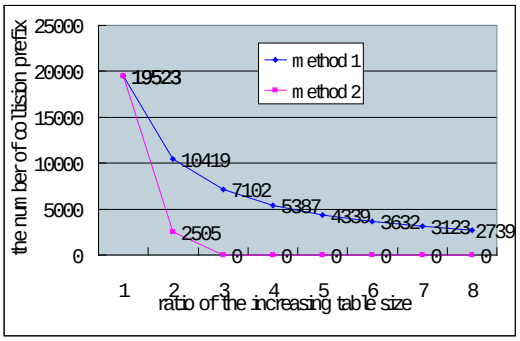

Fig. 4. Simulation result for two methods of increasing hash table size.

as $(108120, 1, n \times 2^{18})$ in method 1, and (k, s, b) is assumed as $(108120, n, 2^{18})$ in method 2. n is defined as ratio of increasing table size. The simulation results are shown in Figure 4. The hash collisions decrease slowly with the first method, but they decrease quickly with the second method. Therefore, the second method is a good choice for hash table.

According to the above simulation results, each hash bucket saves two routes in hash table. The above simulation method is used to generate hash collision information (Table III) for 150000 routing entries. In [9], most active routers have 150000 routing entries, and we design our proposed routing table with 150000 entries. Table III shows the relation between hash table size and overflow. Table III can provide the reference of the CAM size and the collision probability for designer. For example, if hash32's table size is 2^{17}, hash collision probability is 7.63%, and hash collision number is 8256. If the overflow is below 5% and the CAM size is not too big, the Hash32's hash index size should be 18, the Hash48's should be 15, and the Hash64's should be 14 respectively.

C. Solution of Hash Collision

In our proposed hash scheme, CAM is used to solve the hash collision problem. When the hash collision occurs, the new entry is inserted into CAM. CAM deals with the parallel search by searching context and getting the matching result in one search time. Therefore, our hash scheme doesn't need more hash function and more searching time when the collision occurs.

IV. Cache Replacement Algorithm

Figure 5 shows the cache architecture. When cache misses, the new referenced data needs to be inserted into the cache,

TABLE III

MAXIMUM NUMBERS OF COLLISION IN HASH32, HASH48, AND HASH64.

hash table	prefix length distribution	numbers of entry	hash table size	overflow	overflow (%)
Hash32	72.08%	108120	2^{17}	8256	7.63%
			2^{18}	2505	2.32%
			2^{19}	692	0.64%
Hash48	13.15%	19725	2^{15}	890	4.51%
			2^{16}	257	1.30%
			2^{17}	69	0.35%
Hash64	6.66%	9990	2^{13}	1396	13.98%
			2^{14}	460	4.61%
			2^{15}	133	1.33%

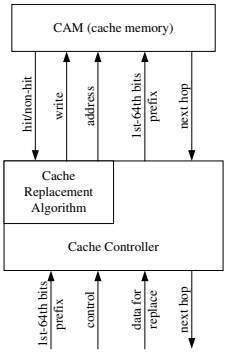

Fig. 5. Cache architecture.

and the insert method is named cache replacement algorithm. A simulation analyzing five cache replacement algorithm's performance with NLNAR's traffic [13] was finished. These five algorithms are FIFO [14], LRU [14], mLRU [15], SF-LRU [16], and LFU [17]. The simulation result is shown in Table IV.

In the simulation, SF-LRU and LFU have great performance. SF-LRU needs counters to record the last reference time and sorting action that is too complex in hardware design. LFU also needs counters to record the last reference frequency. Counter size depends on cache size, and it is additional cost of chip size. FIFO has good performance in network traffic, and it only needs one register to record next CAM address for replacement. So FIFO is a good solution for routing lookup hardware design.

TABLE IV

SIMULATION RESULT OF CACHE REPLACEMENT ALGORITHMS.

number of cache entries	LRU	mLRU (4 segments)	mLRU (16 segments)	SF-LRU	LFU	FIFO
16	44.41%	36.20%	20.68%	44.81%	38.10%	37.86%
32	49.37%	42.21%	27.07%	49.66%	42.86%	43.28%
64	54.72%	45.84%	36.20%	57.15%	48.65%	49.21%
128	61.50%	49.91%	42.21%	64.10%	55.71%	56.07%
256	67.27%	57.33%	45.84%	71.50%	63.50%	63.46%
512	73.52%	64.10%	49.91%	79.66%	71.97%	72.06%
1024	79.78%	71.50%	57.33%	87.32%	85.76%	81.19%

TABLE V

ROUTING LOOKUP SPEED IN IDEAL CASE.

cache hit ratio	clock period on chip	clock period off chip	average lookup time	lookup speed	provide line rate (64B)
50%	3.3ns	3.3ns	6.765ns	147.82Mlps	75.68Gb/s
		5ns	7.7ns	129.87Mlps	66.49Gb/s
		10ns	10.45ns	95.69Mlps	49Gb/s
60%	3.3ns	3.3ns	6.072ns	164.69Mlps	84.32Gb/s
		5ns	6.82ns	146.63Mlps	75.07Gb/s
		10ns	9.02ns	110.86Mlps	56.76Gb/s
70%	3.3ns	3.3ns	5.379ns	185.91Mlps	95.19Gb/s
		5ns	5.94ns	168.35Mlps	86.2Gb/s
		10ns	7.59ns	131.75Mlps	67.46Gb/s
80%	3.3ns	3.3ns	4.686ns	213.4Mlps	109.26Gb/s
		5ns	5.06ns	197.63Mlps	101.19Gb/s
		10ns	6.16ns	162.34Mlps	83.12Gb/s

V. EFFICIENCY ANALYSIS

The fastest line rate is 40Gb/s (OC-768) now, and the smallest frame size of Ethernet frame is 64 Bytes. Therefore, the router should deal with 78.125×10^6 packets per second. According to the prefix length distribution, we assume that 90% search actions can be finished in first clock, and 10% search actions are finished in second clock.

Today, the clock period of commercial CAM is 10ns, and that of commercial SRAM can approach 5ns. Besides the commercial CAM and SRAM speed, the implemented ASIC's clock period is 2.5ns. CCUEE SOC lab proposes a PF-CDPD CAM [18], and its clock period can approach 3.3ns when CAM size is $1024 \times (64 + 8)$. The above information can be used to calculate the routing lookup speed. We assume the ideal off-chip clock period can approach the ASIC's speed, i.e. 3.3ns. We refer to the actual SRAM clock period (5ns) and CAM clock period (10ns). Finally, the analysis results of lookup speed in ideal case is shown Table V.

In Table V, lookup speeds can satisfy the OC-768 requirement. The best ideal lookup speed is 213.4Mlps with 80% hit ratio.

VI. CONCLUSION

The proposed IPv6 routing lookup system is composed of routing lookup ASIC and memory set. The scheme is based on the prefix length distribution of 6Net router's routing table, and the first level of proposed routing table covers about 91.89% routing entries. Consequently, the most routing lookup can be searched in first memory access, and the worst case is two memory accesses. The ASIC is composed of a function unit and a CAM. The function unit does insert, search, update and delete actions in routing lookup system. The CAM is used as cache memory and records 64-bits prefix and next hop information. Additionally, FIFO is used as cache replacement algorithm in the proposed architecture. The CAM with 1024 entries can guarantee 80% hit ratio, and the routing lookup speed can approach 213.4Mlps and satisfies the requirement of OC-768. It only needs 20.04KB TCAM, 10.24KB BCAM, and 29.29MB RAM for 150000 routing entries. Table VI shows the comparison with the related routing lookup research.

TABLE VI

COMPARISON.

	Our Scheme	BDD [3]	IPv4/IPv6 dual [7]	TCAM for Network Application [6]
implement method	hardware	software	hardware (TCAM)	hardware (TCAM)
worst search latency	2 memory access	depend on trie depth	7-stage pipeline	3 clock cycle latency, 1 clock is 5ns
lookup speed	213.4Mlps	168.6Mlps for 29487 prefixes	100Mlps	200Mlps
memory size	TCAM: 20.04KB, CAM: 10.24KB, RAM: 29.29MB (150000 prefixes)	non-available	non-available	21Mb capacity, 21632 entries

REFERENCES

[1] Y. Rekhter and T. Li, *RFC1518: An Architecture for IP Address Allocation with CIDR*. Internet Engineering Task Force, Sept. 1993.

[2] R. Hinden, S. Deering, and E. Nordmark, *RFC3587: IPv6 Global Unicast Address Format*. Internet Engineering Task Force, Aug. 2003.

[3] R. Sangireddy and A. Somani, "High-speed IP routing with binary decision diagrams based hardware address lookup engine," *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 513 – 521, May 2003.

[4] B. Lampson, V. Srinivasan, and G. Varghese, "IP lookups using multiway and multicolumn search," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 324 – 334, Jun. 1999.

[5] N.-F. Huang and S.-M. Zhao, "A novel IP-routing lookup scheme and hardware architecture for multigigabit switching routers," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1093 – 1104, 1999.

[6] B. Gamache, Z. Pfeffer, and S. Khatri, "A fast ternary cam design for IP networking applications," in *ICCCN 2003*, 20-22 Oct. 2003.

[7] Z.-X. Wang, H.-M. Wang, and Y.-M. Sun, "High-performance IPv4/IPv6 dual-stack routing lookup," in *18th International Conference on Advanced Information Networking and Applications*, vol. 1, 2004.

[8] T. Hayashi and T. Miyazaki, "High-speed table lookup engine for IPv6 longest prefix match," in *Global Telecommunications Conference, 1999. GLOBECOM '99*, vol. 2, 1999.

[9] Potaroo.net. [Online]. Available: http://bgp.potaroo.net/

[10] R. Jain, "A comparison of hashing schemes for address lookup in computer networks," *IEEE Transactions on Communications*, vol. 40, no. 3, pp. 1570–1573, Oct. 1992.

[11] The computer center of the ministry of education. [Online]. Available: http://www.edu.tw/tanet/introduction.html(inChinese)

[12] M. Ramakrishna, "An exact probability model for finite hash table," in *Fourth International Conference on Data Engineering*, 1-5 Feb. 1991.

[13] Nlnar measurement and network analysis, passive measurement and analysis. [Online]. Available: http://pma.nlanr.net/

[14] A. S. Tanenbaum and A. S. Woodhull, *Operating Systems: Design And Implementation, Second Edition*. Prentice Hall, Dec. 1996.

[15] H. Liu, "Reducing cache miss ratio for routing prefix cache," in *Global Telecommunications Conference (IEEE GLOBECOM 2002)*, vol. 3, 17-21 Nov. 2002.

[16] J. Alghazo, A. Akaaboune, and N. Botros, "SF-LRU cache replacement algorithm," in *Records of the International Workshop on Memory Technology, Design and Testing (MTDT'04)*, 9-10 Aug. 2004.

[17] W.-L. Shyu, C.-S. Wu, and T.-C. Hou, "Efficiency analyses on routing cache replacement algorithms," in *IEEE International Conference on Communications (ICC 2002)*, vol. 4, 28 April - 2 May 2002.

[18] J.-S. Wang, H.-Y. Li, C.-C. Chen, and C. Yeh, "An and-type match-line scheme for energy-efficient content addressable memories," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 9 Feb. 2005.