

# Hybrid Memory Design for High-Throughput and Low-Power Table Lookup in Internet Routers

Hayato Yamaki

Graduate School of Informatics and Engineering, The University of Electro-Communications  
1-5-1 Chofugaoka, Chofu 182-8585, Japan

**Abstract**—Table lookup is a major process to decide the packet processing throughput and power efficiency of routers. To realize high-throughput and low-power table lookup, recent routers have employed several table lookup approaches, such as TCAM (Ternary Content Addressable Memory) based approach and DRAM (Dynamic Random Access Memory) based approach, depending on the purpose. However, it is difficult to realize both ultrahigh throughput and significant low power due to the trade-off between them. To satisfy both of the demands, this study proposes a hybrid memory design, which combines TCAM, DRAM, PPC (Packet Processing Cache), CMH (Cache Miss Handler), and IP Cache, to enable a high-throughput and low-power table lookup. The simulation results using an in-house cycle-accurate simulator showed that the proposed memory design achieved nearly 1 Tbps throughput with similar power of the DRAM-based approach. When compared to the approach proposed in a recent study, the proposed memory design can realize 1.95x higher throughput with 11% power consumption.

**Keywords**—Inter routers; packet processing; table lookup; hybrid memory architecture; Packet Processing Cache (PPC)

## I. INTRODUCTION

A demand for high-throughput and low-power packet processing is becoming serious in routers year by year due to an increase in internet traffic. For example, to achieve 400 Gbps, routers must process packets every 1.28 nano second if the shortest packets arrive continuously. Moreover, because the power consumption of routers increases depending on the number of packets processed, the power efficiency is also an important factor for recent routers. According to the reports [1]–[4], the total power consumed by the network devices will reach several percentages of the total power generated in the world. Thus, routers must process packets at high throughput with low power.

Table lookup is a key process to decide the packet processing throughput and power efficiency of routers. When a packet arrives at a router, the router searches tables, such as a routing table and access control list, to decide the next-hop IP address and whether filtering. Because this process requires memory accesses per packet, table lookup is a bottleneck of the packet processing throughput and a power-hungry operation in routers.

To increase the throughput or reduce the power consumption, several table lookup approaches have been employed. A DRAM (dynamic random access memory) based approach is the most standard approach of the table lookup [5], [6]. This approach stores the tables into DRAM, which can be implemented with a large capacity at low expense and low power. Thus, the DRAM-based approach has been employed

in enterprise edge routers, which are required to be introduced at low expense, and application routers, which process packets with fine grained services. However, because the access latency of DRAM is large, this approach cannot satisfy the demand of ultrahigh throughput, such as 400 Gbps and 1 Tbps.

In contrast to the DRAM-based approach, TCAM (ternary content addressable memory) based approach has been used for high-throughput table lookup [7]–[10]. The TCAM is a specialized memory for fast data lookup and can retrieve data at one cycle. Because the access latency of TCAM is smaller than that of a DRAM, this approach can achieve larger throughput than the DRAM-based approach. Thus, the TCAM-based approach has been employed in core routers and data-center routers, which are required to process packets at high throughput. However, to retrieve data at one cycle, TCAM consumes significant large power in comparison to DRAM. Accordingly, it was reported that TCAM consumed 40% of the total power consumed in a router [11]. The paper [9] also indicated that TCAM consumed 150 times larger power than same-sized DRAM.

Due to the trade-off between the access latency and power consumption of memories, as mentioned above, satisfying the demands of both throughput and power efficiency is difficult for these approaches. In this study, a novel hybrid memory design, which combines the DRAM- and TCAM-based approaches and further adds PPC (packet processing cache), CMH (cache miss handler), and IP Cache, is proposed for high-throughput and low-power table lookup in routers.

This study also builds an in-house cycle-accurate table lookup simulator which can simulate not only the proposed memory design but also the other conventional table lookup approaches (e.g., the DRAM-based or TCAM-based approaches). This simulator enables to measure the table lookup throughput of a router considering the hardware behavior (e.g., stalling and queuing) while most previous studies evaluated the throughput based on a mathematical analysis using a throughput model without considering the concrete hardware. Thus, this study newly reveals the impact of the hardware constraints (i.e., memory stalls, hash conflicts, and the number of buffer entries) on the table lookup throughput.

The major contributions of this study are summarized below:

- This study proposed a novel hybrid memory design for high-throughput and low-power table lookup in routers. The simulation results showed the proposed memory design can realize nearly 1 Tbps throughput with similar power of the DRAM-based approach.

- This study revealed the concrete hardware design of various table lookup approaches and their connection. Most previous studies did not consider the concrete hardware design of them, and thus, the impact of the hardware design on the table lookup performance was not revealed.
- This study is a first study of evaluating the throughput of the table lookup with caches considering the hardware constraints (i.e., memory stalls, hash conflict, and the number of buffer entries) by measuring them based on cycle-accurate simulations.

The remainder of this paper is organized as follows: Section 2 describes the table lookup process in a router. Section 3 introduces details of table lookup approaches used in this study as related works. We propose the efficient memory design for the high-throughput and low-energy table lookup in Section 4 and evaluate it in Section 5. Finally, Section 6 concludes this paper.

## II. TABLE LOOKUP

Routers have several tables to correctly decide how to process packets. In general, typical routers have four tables: a routing table, ARP (address resolution protocol) table, ACL (access control list), and QoS (quality of service) table. Note that several router vendors have different named tables, such as a FIB and adjacency table in Cisco routers, instead of the routing table and ARP table. However, they make no difference in this study, and thus, we suppose that routers have the four tables mentioned above in this study hereafter.

Fig. 1 illustrates a basic flow of packet processing in a router. When a packet arrives at a router, the whole data of the packet is stored to a packet memory, and only the header part is sent to next processing flow (i.e., a table lookup module). It is because the payload is not required to process the packet. The table lookup module searches the tables using the packet header information. For example, the routing table lookup is done using a destination IP address while the ACL lookup is done using the five-tuple (i.e., source and destination IP addresses, source and destination port numbers, and protocol number). After searching the tables, the router recalculates the ttl (time to live) and checksum value from the packet header. Based on the table lookup results and recalculation values, the router modifies the packet header and concatenates the modified header with the packet payload read from the packet memory. Finally, the packet is forwarded to a next hop.

In the packet processing, the table lookup is a major throughput bottleneck due to the large access latency. As it is often discussed, memory-wall, which represents the large gap between CPU operation frequency and memory access latency, is the most serious problem of computer architecture, and it is also a problem in routers. Make matters worse, each table requires a large number of entries. For example, all BGP entries in the routing table exceeded 1M entries. Consequently, the table lookup conventionally takes from dozens to a hundred cycles although other processes (i.e., the calculation and header modification) take several cycles. Thus, improving the table lookup is the most important issue to increase the packet processing throughput.

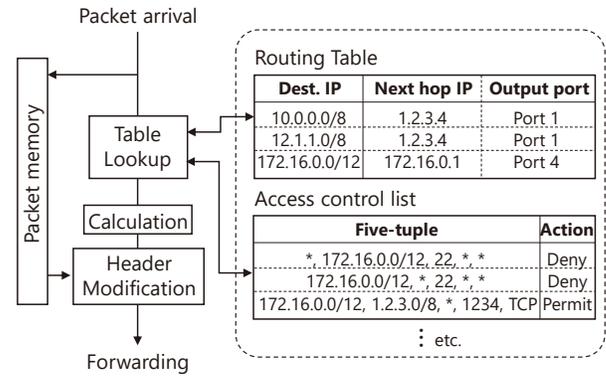


Fig. 1. Outline of table lookup in a router.

TABLE I. SUMMARY OF EACH APPROACH.

Approach	Throughput	Energy	Notes
<b>DRAM only</b> (e.g., [5], [6])	Low	Low	Low expense
<b>TCAM only</b> (e.g., [7]–[10])	High	High	Lookup at one cycle
<b>IP Cache + DRAM</b> (e.g., [12]–[15])	Middle	<b>Very low</b>	Performance depends on the cache miss rate
<b>PPC + TCAM</b> (e.g., [16]–[19])	Very high	Middle	Same as above
<b>PPC + CMH + TCAM</b> (e.g., [20], [21])	<b>Very high+</b>	Middle	Non-blocking

## III. RELATED WORKS

As explained in Section 2, the table lookup is the most important factor for routers to determine the packet processing throughput. Consequently, many studies are conducted to improve the table lookup. This section introduces major approaches which have been employed in routers. Table I summarized characteristics of each approach.

### A. DRAM-based Approach

A DRAM-based approach is the most standard approach for the table lookup [5], [6]. In this approach, the tables are stored into DRAM. The DRAM is a typical memory and can be implemented with a large capacity at low expense. Thus, this approach can realize the table lookup at lowest expense compared to other approaches.

Conventionally, there are several methods to find data in DRAM. The most simple method is to search entries from the head to the end by linear search until the data are found. However, this method significantly increases the DRAM accesses because the tables require a large number of entries, as mentioned in Section 2. Using hash values as addresses of data in DRAM is another method. However, this way causes conflicts of addressing data.

Tree-based methods, such as Radix trees and Patricia trees, are the most standard methods for the table lookup in routers [5], [6]. Fig. 2 depicts the structure of the Radix tree as a representative. In the tree-based methods, binary trees are traced based on each bit of the destination IP address. Conclusive stopped nodes indicate the routes for the destination

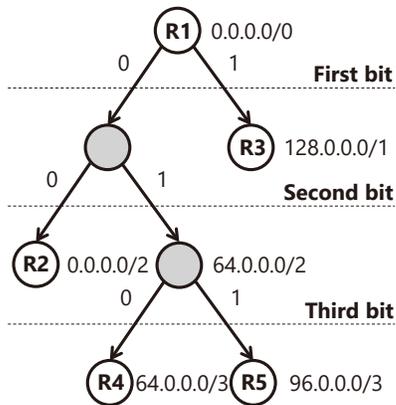


Fig. 2. Structure of Radix tree.

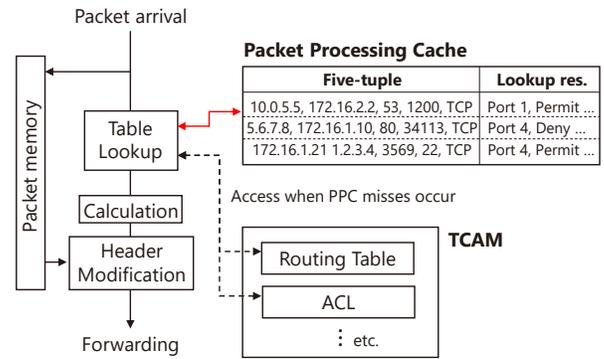


Fig. 3. Outline of table lookup with PPC.

IP address. This tracing process can be done through pipeline and thus, achieved high throughput.

A problem of the DRAM-based approach results from the large delay of DRAM accesses. The DRAM takes a long time to retrieve data due to the DRAM structure. Although routers must process packets each several nano seconds to achieve 400+ Gbps, as mentioned in Section 1, approximately 50 nano seconds are required for a DRAM access. The DRAM-based approach is comparatively slow in comparison to other approaches, while it can process packets with significant low power.

### B. TCAM-based Approach

A TCAM-based approach is employed in routers which require high throughput, such as core routers [7]–[10]. This approach stored the tables into TCAM. The TCAM is a specialized memory for fast data lookup and can search data at one cycle. Because the access latency of TCAM is also small (approximately several nano seconds), the TCAM-based approach can achieve high throughput.

A problem of the TCAM-based approach is that TCAM consumes significant large power for data lookup. It is because TCAM compares all stored bits simultaneously to obtain data at one cycle. The papers [8] reported that TCAM consumes 150 times larger energy than a same sized RAM. Therefore, this approach achieves high throughput at the sacrifice of power consumption. According to the papers [11], TCAM consumes 40% of total power consumed in a router with TCAM.

### C. IP Cache

IP Cache is a supplemental approach of the DRAM- and TCAM-based approaches and accelerates the table lookup with reducing the power consumption [12]–[15]. IP Cache is placed before accessing the DRAM or TCAM. It stores table lookup results of the routing table and ARP table per destination IP address into a small SRAM (static RAM) and references them to process subsequent packets. Because the packets of the same destination IP address indicate the same routing table lookup result and ARP table lookup result, IP Cache can reduce the

number of DRAM or TCAM accesses if IP Cache has the table lookup results of the packets.

A small SRAM used as IP Cache shows low access latency and low power consumption. For example, the access latency and energy consumption of a 32KB SRAM, which is conventionally used in IP Cache, are 0.5 ns and 0.0539 nJ per access, respectively, while those of TCAM are 5 ns and 30 nJ per access, respectively [18]. Thus, it is important for IP Cache to process packets using only SRAM as many as possible.

The entries of IP Cache are constructed by 4 bytes tag (i.e., a destination IP address) and 6+ bytes data (i.e., table lookup results of the routing table and ARP table). Thus, a typical 32KB IP Cache has approximately 3K entries. Conventionally, IP Cache is configured by a 4-way set associative cache, and the entries are mapped using the CRC hash values calculated from destination IP addresses. The paper [22] showed that a 32KB 4-way IP Cache can achieve cache hit rates from 80% to 90%. Moreover, IP Cache has a possibility to achieve the cache hit rate of up to 98% when increasing the capacity. It indicates that IP Cache has possibility to process most packets with only a SRAM.

The throughput and power consumption of the table lookup produced by IP Cache mainly depend on two factors: the cache hit rate and cache performance (i.e., the access latency and power consumption). The cache hit rate is the most important factor to determine the table lookup performance because it represents the rate of packets processed using the SRAM. To achieve a high cache hit rate, increasing the SRAM capacity is the most simple solution. However, it is not suitable considering the SRAM performance. For example, the latency of a 1MB SRAM is almost the same as that of TCAM. Thus, achieving a high cache hit rate with small capacity SRAM is required to exhibit further performance.

### D. Packet Processing Cache

Similar to IP Cache, PPC has been proposed as a supplemental approach of mainly the TCAM-based approach to realize both accelerating the table lookup and reducing the power consumption [16]–[18]. PPC solves the problem of the TCAM-based approach that TCAM consumes large power. Unlike IP Cache, PPC caches lookup results of all kinds of tables in a router (i.e., four tables) per flow into a SRAM. If

PPC has the table lookup results of a flow, subsequent packets of the same flow can be processed without accessing TCAM. Consequently, PPC can substitute one PPC access for four TCAM accesses while IP Cache substitutes one cache access for two TCAM accesses (i.e., the routing table access and ARP table access).

Fig. 3 shows the outline of the table lookup with PPC. PPC stores table lookup results per five-tuple (i.e., source and destination IP addresses, source and destination port numbers, and protocol number), which is called a flow in PPC. It is because most tables in routers store data based on the some or all the five-tuple. PPC can process subsequent packets of flows using only PPC by caching the table lookup results of the first packets of the flows into PPC. In comparison to IP Cache, although flows have a smaller temporal locality than packets of the same destination IP address, PPC can process packets with one PPC access. Thus, PPC has a possibility to further improve the throughput and power consumption of the table lookup in comparison to IP Cache if it achieves a high PPC hit rate.

By combining PPC with the TCAM-based approach, it can achieve ultrahigh table lookup throughput. However, the power consumption is still high due to remaining TCAM accesses. The paper [19] indicated that the dynamic energy of TCAM was still dominant in a router even if PPC achieves the PPC hit rate of 95%.

The entries of PPC are constructed by 13 bytes tag (i.e., five-tuple) and 15+ bytes data (i.e., table lookup results of the four tables). Thus, a typical 32KB PPC has approximately 1K entries. Similar to IP Cache, conventional PPC is configured by a 4-way set associative cache, and the entries are mapped using the CRC hash values calculated from five-tuples. Because the PPC entry size is larger than the IP Cache entry size, the PPC hit rate tends to become lower than IP Cache. A 32KB PPC shows the PPC hit rate of approximately 70% [23].

#### E. Cache Miss Handler

CMH was proposed in [20], [21] to assist PPC and enable to process packets without blocking. When a packet of a flow misses in PPC, the router must block the table lookup operation of subsequent packets of the same flow until the entry of the flow is prepared in PPC. It is because subsequent packets may continuously miss in PPC before completing the table lookup of the flow and updating it in PPC. Thus, CMH queues subsequent packets of the flow until the PPC update of a former packet of the flow is completed.

Fig. 4 illustrates the overview of CMH. It is placed between PPC and TCAM and accessed if packets miss in PPC. CMH is composed of two modules: CMT (cache miss table) and CMQ (cache miss queues).

CMT is implemented by a small full-associative cache and manages the flows which are being processed in TCAM. When a packet missed in CMT, CMT stores the five-tuple and set the valid bit on if a CMT entry is empty. The subsequent packets with the same five-tuple are sent to CMQ, which is simple FIFOs, by hitting in CMT. At this time, the queue number of CMQ is determined based on the hit address of CMT. After the CMT miss packet is processed in TCAM, the release signal

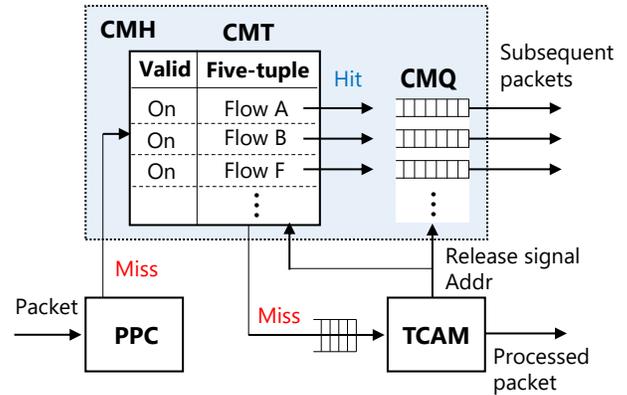


Fig. 4. Overview of CMH.

and CMT address (i.e., the queue number) are sent to CMT and CMQ. Finally, the CMT entry is disabled by setting the valid bit off, and the packets queued in CMQ are released with the TCAM lookup results.

#### IV. HYBRID MEMORY DESIGN FOR HIGH-THROUGHPUT AND LOW-ENERGY TABLE LOOKUP

As explained in Section 3 and summarized in Table I, each table lookup approach contributes to increasing the throughput or reducing the power consumption. However, there are no approaches to realize ultrahigh throughput at significant low power consumption. For example, the combination of PPC, CMH, and TCAM shows the highest throughput in the approaches shown in Table I. However, it still consumes large power due to the remaining TCAM accesses. Our targets of the throughput and power consumption are shown as bold in Table I. To meet these requirements, this study proposes a novel hybrid memory design for high-throughput and low-power table lookup in routers.

Fig. 5 depicts an overview of the proposed memory design. In this design, the five approaches introduced in Section 3 are combined using four buffers, called PPC buffer, Victim buffer, Table buffer, and DRAM buffer, in suitable order. Because the latter memory lookup is slower in the proposed memory design, the buffers are required. Each buffer enables to operate the former memory lookup independently of the latter memory lookup. If a buffer is filled, the former memory lookup is stopped until the buffer becomes available. Details of each combination are explained hereinafter.

1) *Combination of PPC and CMH*: The combination of PPC and CMH was already considered in [20], [21], and thus, the processing follows these papers. Packets missed in PPC are sent to CMH and judged whether hit or missed in CMT. When a packet hits in CMT, the packet is sent to CMQ and queued until the former packet of the same flow is processed by DRAM or TCAM. On the other hand, when a packet misses in CMT, the packet is sent to IP Cache, and the new CMT entry is registered, as mentioned in Section 3.

In the proposed memory design, the number of CMT

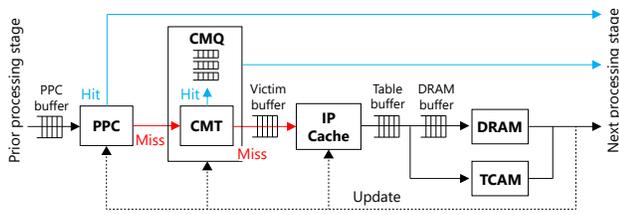


Fig. 5. Overview of proposed memory design.

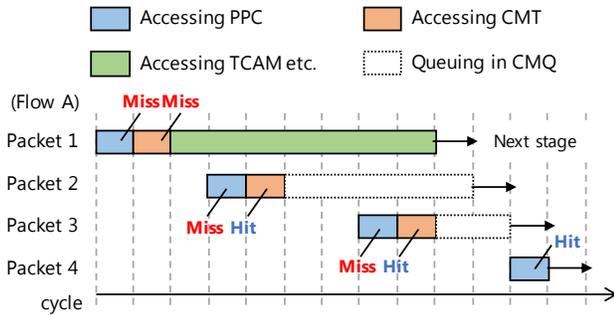


Fig. 6. Time chart of processing with our memory design.

entries becomes the same as the summation of the number of buffer entries of Victim buffer, Table buffer, and DRAM buffer to manage all flows being processed in the latter memories. Larger buffer sizes enable to process a larger number of packets without blocking and achieve higher table lookup throughput. However, there is a trade-off between the buffer sizes and the implementation costs of the buffers, CMT, and CMQ. Consequently, deciding the number of CMT entries and buffer sizes considering both the throughput and implementation costs is important.

We newly discuss the behavior of the combination of PPC and CMH. First, CMH guarantees the order of packets processed in the proposed memory design at flow level. Fig. 6 shows this behavior using a time chart of the processing. Packets of the same flow are sent to the next processing stage in the arrival order, and out of order due to the latency gap between PPC and DRAM or TCAM does not occur in this memory design. It is because CMH keeps subsequent packets of a flow waiting until the former packet of the flow is processed in DRAM or TCAM. Second, CMH does not require a buffer between PPC and CMT. It is because the CMT access is always faster than PPC due to the small capacity.

2) *Combination of CMH and IP Cache:* The combination of PPC and IP Cache was also considered in [22]. In this study, IP Cache is placed after CMH, and packets which missed in CMT are sent to IP Cache after queuing in Victim buffer. Because IP Cache enables to process packets based on the destination IP address, it has a possibility to achieve a significant higher hit rate than PPC. Moreover, the proposed memory design allows IP Cache to make the capacity larger because packets which sent to IP Cache is significantly small in comparison to PPC, and large IP Cache latency is permissible. It also induces an increase in the cache hit rate. Note that

packets are sent to Table buffer regardless of the hits or misses in IP Cache because packets must access DRAM or TCAM even if they hit in IP Cache.

3) *Combination of DRAM and TCAM:* In the proposed memory design, the DRAM and TCAM are combined to reduce the power consumption with increasing the throughput. To meet the requirement of the power efficiency, packets should be processed by the DRAM as many as possible, especially when packet arrival is a slow. The TCAM is used in the case that the DRAM lookup is too late for processing packets.

To realize these behaviors, a simple method using two buffers is proposed in this study. As depicted in Fig. 5, packets which missed in IP Cache are first sent to Lookup buffer. The packets queued in Lookup buffer are next sent to DRAM buffer, which placed before the DRAM, until DRAM buffer is filled up. If DRAM buffer is full, the packets queued in Lookup buffer are sent to the TCAM and processed using the TCAM. Thus, the TCAM are utilized only when the DRAM is busy, and the power consumed by the table lookup can be reduced as large as possible. In this study, we consider that one entry is enough as the DRAM buffer size. It is because increasing the DRAM buffer size causes the increase in CMT and CMQ entry sizes. Moreover, the DRAM buffer size does not significantly impact on the throughput and power consumption.

After a packet is processed using the DRAM or TCAM, the corresponding CMT entry and packets queued in CMQ are released, and the table lookup results are cached into IP Cache and PPC. The processed packet and queued packets are sent to the next processing stage in the arrival order.

## V. EVALUATION

This section shows the evaluation of the proposed memory design. In this evaluation, the table lookup operation in a router is simulated using an in-house cycle-accurate table lookup simulator and packet traces (i.e., pcap files) captured in real networks. The throughput and power consumption of the table lookup can be measured using this simulator. In this study, the evaluation was done based on following points.

- Effect of the combination of DRAM and TCAM
- Comparison to other approaches

### A. Simulation Environment

1) *Cycle-accurate Table Lookup Simulator:* To evaluate the proposed memory design, an in-house table lookup simulator, written in C++, was used. This simulator can simulate the table lookup operation in a router including the queuing and stalling at cycle level. The architecture of the simulator was modeled in Fig. 5.

Table II shows the parameters set in the simulator and the reference values. In the following simulations, the reference values are used if the values are not written clearly. These values were mainly decided based on the previous studies such as [18], [20], [22]. The latency, dynamic energy, and static power of PPC, CMH, IP Cache, and DRAM were estimated using CACTI 7.0 [24], which was a major tool for estimating the latency and power consumption of various

TABLE II. SIMULATOR PARAMETERS AND REFERENCE VALUES.

Item	Reference value
Operation frequency	2 GHz (0.5 ns / cycle)
PPC	
Associativity	4 ways
Entries	1,024 entries
Replacement	LRU
Latency	1 cycle
Dynamic energy	0.0342 nJ
Static power	40 mW
CMH	
Associativity	full associative
CMT entries	32 entries
CMQ entries / queue	10 entries
Latency	1 cycle
Dynamic energy	0.0346 nJ
Static power	4.53 mW
IP Cache	
Associativity	4 ways
Entries	4,096 entries
Replacement	LRU
Latency	1 cycle
Dynamic energy	0.0208 nJ
Static power	12.8 mW
DRAM	
Latency	10 cycles
Number of ports	4 ports
Dynamic energy	1.26 nJ
Static power	13.9 mW
TCAM	
Latency	100 cycles
Number of ports	4 ports
Dynamic energy	166 nJ
Static power	17 mW
Buffers	
PPC buffer	unrestricted
Victim buffer	10 entries
Lookup buffer	10 entries
DRAM buffer	1 entries

TABLE III. SIMULATOR PARAMETERS AND REFERENCE VALUES.

Trace Name	Bandwidth	Number of packets	Duration
WIDE	1 Gbps	22,172,838 packets	900 s
Academic	10 Gbps	2,411,883 packets	90 s

types of memories, while those of TCAM were estimated from recent CAM's studies [25], [26]. We also note that the numbers of ports in a DRAM and TCAM were set to four because they can search the four tables in a router independently.

2) *Packet Traces*: As workloads, two pcap-format packet traces captured in real networks were used. Details of them are summarized in Table III. WIDE (Widely Integrated Distributed Environment) trace contains communication traffic at the 1-Gbps transit link of WIDE to the upstream ISP and can be obtained from [27]. Academic trace contains communication traffic at a 10-Gbps core link in a institute which mixed various University traffic and is not opened to the public. The pcap files include information of a packet per line, namely, the arrival time, each header information (i.e., Ethernet header, IP header, and TCP/UDP headers), and some payload. The simulator can obtain packets by reading each line in the pcap files.

3) *Three Simulation Conditions*: To evaluate the throughput and power consumption based on the practical use, this study conducted simulations under the three conditions: the full load, 400-Gbps load, and 100-Gbps load.

The full-load simulation was conducted to measure the achievable throughput and maximum power consumption. It starts the table lookup simulation without considering the arrival time of packets. It means that the simulation is started under the situation that all packets are queued in PPC buffer. In routers, the throughput in the situation that packets are queued in buffers is the most important because packets are dropped if

TABLE IV. COMPARISON OF TABLE LOOKUP PERFORMANCE WITH THREE DIFFERENT DESIGNS.

		DRAM-only	TCAM-only	Proposed
WIDE	Throughput	65 Gbps	660 Gbps	727 Gbps
	Power (100-Gbps)	89.9 mW	377 mW	142 mW
	(400-Gbps) (Full-load)	124 mW 152 mW	1,430 mW 33,700 mW	386 mW 33,800 mW
Academic	Throughput	84 Gbps	854 Gbps	929 Gbps
	Power (100-Gbps)	108 mW	936 mW	166 mW
	(400-Gbps) (Full-load)	189 mW 153 mW	3,560 mW 33,700 mW	1,390 mW 33,400 mW

the throughput is insufficient. Note that this study measured the throughput on the assumption that all packets were constructed of 64 bytes (i.e., the shortest packet length). It is because routers conventionally show this worst-case throughput as an important barometer of the packet processing throughput.

On the other hand, the 400- and 100-Gbps simulations were also conducted to measure the power consumption under the specific traffic loads. In these simulations, the arrival times of each packet were modified to satisfy the bandwidths of 400- or 100-Gbps considering the packet length, and packets were sent to the simulator in accordance with the modified arrival time. Note that the throughput measured in these simulations was not meaningful because the systems obviously had the capability to achieve 400-Gbps throughput (not the shortest-packet-length throughput). Beside the throughput, the power consumption measured in these simulations is important because routers are not always operating under a full load. Consuming the power under the full-load condition is rare for routers.

### B. Effect of the Combination of DRAM and TCAM

First, effect of the combination of the DRAM and TCAM was evaluated. To reveal it, this study implemented the approaches that all packets which missed in IP Cache were assigned to only the DRAM or TCAM (referred to as DRAM-only design and TCAM-only design, respectively) for comparison.

Table IV summarizes the throughput and power consumption of the table lookup measured in the simulations. As shown in the table, the proposed memory design achieved significant higher throughput than the DRAM-only design, and it slightly overcome the TCAM-only design because the proposed memory design can process packets using both the DRAM and TCAM. According to the full-load simulations, the proposed memory design achieved 11.0x and 1.09x higher throughput than the DRAM-only and TCAM-only designs, respectively, on average. In Academic trace, the proposed memory design achieved nearly 1-Tbps throughput, and this result substantiated the previous studies which analyzed the throughput based on the mathematical model.

Table IV also shows usefulness of the proposed memory design from the aspect of the power consumption. In the full-load condition, the proposed memory design showed almost the same power of the TCAM-only design because most packets were assigned to the TCAM to speed up the table lookup. However, in the 100- and 400-Gbps conditions, the proposed memory design can significantly reduce the power in comparison to the TCAM-only design. The results showed

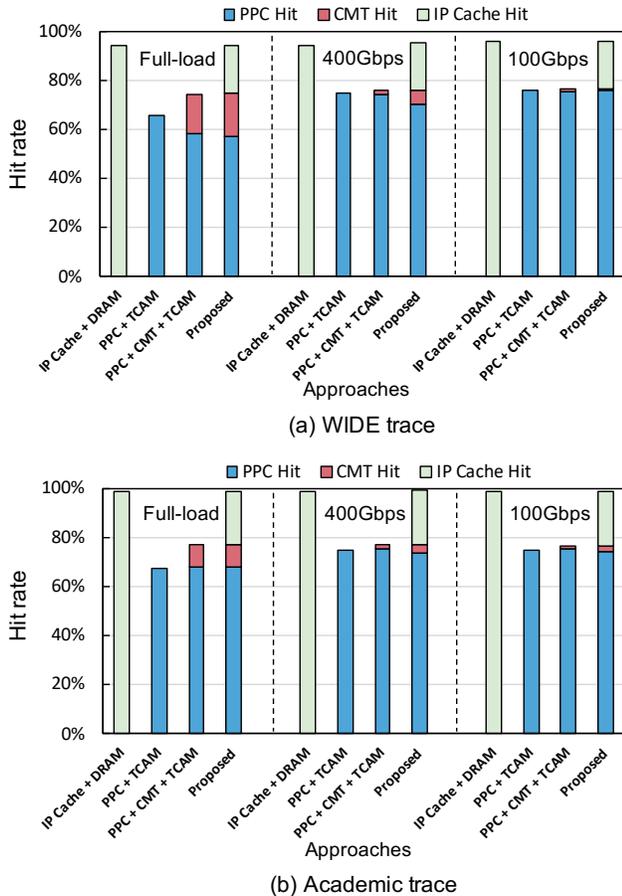


Fig. 7. Comparison of hit rates and the breakdown.

that the proposed memory design can reduce the power consumption by 72.3% and 67.0% in the 100- and 400-Gbps conditions, respectively, in comparison to the TCAM-only design. These results indicate that the proposed memory design can significantly reduce the power consumption of the table lookup with keeping the throughput to the same level as the TCAM-only design.

### C. Comparison to Other Approaches

This section reveals superiority of the proposed memory design to other approaches from the perspectives of the throughput and power efficiency. For comparison, this study implemented five conventional approaches: the TCAM-based approach, DRAM-based approach, combination of IP Cache and DRAM [12], combination of PPC and TCAM [16], and combination of PPC, CMT, and TCAM [23].

First, the cache hit rates achieved by each approach were evaluated. Fig. 7 showed the cache hit rates and their breakdown under the three load conditions. According to the results, the proposed memory design can achieve significant high hit rate (94% in WIDE trace and 99% in Academic trace) by combining PPC and IP Cache. The combination of IP Cache and DRAM also achieved high hit rate; however, it did not significantly impact on the table lookup performance because it required DRAM accesses even if packets hit in IP Cache.

In addition, the larger the load of a router increased, the more CMH assisted the cache hit rate. This is because the number of packets still waiting to be processed by the DRAM or TCAM becomes large when the load of a router increases. If a router does not employ CMH (i.e., in the case of *PPC + TCAM* in Fig. 7), this situation causes a large number of PPC misses because it takes time to update PPC entries. This behavior also reveals the reason that the PPC hit rate of the proposed memory design was a little lower than that of the *PPC + CMT + TCAM*. In the proposed memory design, there is a possibility that packets are waited for a longer time compared to TCAM-based approaches because packets may assign to the DRAM. Thus, the PPC hit rate may become a little lower; however, it is no problem for the table lookup performance because it can be saved by CMH, as shown in Fig. 7.

Second, the table lookup throughput and power efficiency were evaluated. Table V summarized them. In the table, the parenthesis values represent the power efficiency calculated from the power consumption divided by the throughput [mW/Gbps], which were often used in routers as a barometer of the router performance. The smaller power efficiency is more suitable for routers; however, considering not only this power efficiency but also the achievable throughput is important for routers.

As shown in Table V, the proposed memory design achieved 1.95x throughput on average compared to *PPC + CMT + TCAM*, which showed the largest throughput in recent studies. It realized up to nearly 1-Tbps table lookup throughput. The proposed memory design also had the advantage of the power efficiency in comparison to other TCAM-used approaches. It reduced the power consumption per Gbps by 86%, 89%, and 44% in 100-Gbps, 400-Gbps, and full-load conditions, respectively. In the full-load condition, the proposed memory design consumes the power as the same level of the other TCAM-used approaches because most packets are assigned to the TCAM. However, the power efficiency is better due to the high achievable throughput. In addition, the full-load condition is rare in routers in practical use. Consequently, it was showed that the proposed memory design can achieve significant high table lookup throughput with low power consumption as the same level of the DRAM-based approach.

## VI. CONCLUSION

The table lookup is the most important operation in routers to determine the packet processing throughput and power consumption. Thus various approaches, such as DRAM-based approaches, TCAM-based approaches, and cache-based approaches, have been proposed and employed. However, there are no approaches to satisfy the requirements of both ultrahigh throughput and significant low power consumption.

For realizing them, this study proposed a novel hybrid memory design, which combines five conventional approaches (i.e., PPC, CMH, IP Cache, DRAM, and TCAM) in the appropriate order. The effectiveness of the proposed memory design was evaluated using an in-house simulator which can simulate the table lookup in a router at cycle level. The simulation results indicated that the proposed memory design achieved

TABLE V. COMPARISON OF TABLE LOOKUP PERFORMANCE WITH VARIOUS CONVENTIONAL APPROACHES.

		DRAM-based	TCAM-based	IPCache+DRAM	PPC+TCAM	PPC+CMT+TCAM	Proposed
WIDE	Throughput	10 Gbps	102 Gbps	19 Gbps	297 Gbps	401 Gbps	727 Gbps
	Power (100-Gbps)	91 mW	2,550 mW	41 mW	651 mW	638 mW	95 mW
		(12)	(333)	(5)	(85)	(83)	(12)
	(400-Gbps)	115 mW	10,320 mW	103 mW	2,640 mW	2,550 mW	287 mW
		(11)	(329)	(5)	(84)	(81)	(9)
	(Full-load)	115 mW	33,600 mW	103 mW	33,300 mW	33,700 mW	33,800 mW
		(11)	(328)	(5)	(112)	(84)	(47)
Academic	Throughput	10 Gbps	102 Gbps	20 Gbps	313 Gbps	444 Gbps	930 Gbps
	Power (100-Gbps)	115 mW	6,900 mW	103 mW	1,770 mW	1,660 mW	145 mW
		(11)	(330)	(5)	(84)	(80)	(7)
	(400-Gbps)	115 mW	29,800 mW	103 mW	7,510 mW	6,780 mW	1,407 mW
		(11)	(329)	(5)	(83)	(75)	(16)
	(Full-load)	115 mW	33,600 mW	103 mW	33,700 mW	33,700 mW	33,400 mW
		(11)	(328)	(5)	(108)	(76)	(36)

nearly 1 Tbps throughput with similar power consumption of the DRAM-based approach. If compared to the table lookup approach with PPC, CMT, and TCAM, which shows the largest throughput in recent studies, the proposed memory design achieved 1.95x throughput with 11% power consumption at 400-Gbps condition.

As future works, the increase in the PPC hit rate is one of the important issues to further improvement in the table lookup performance. Although the proposed memory design shows a high cache hit rate (more than 95%), it does not largely impact on the performance due to the remaining DRAM and TCAM accesses. To reduce the number of DRAM and TCAM accesses, improving the PPC hit rate is the most effective approach. Thus, more efficient PPC design should be considered.

#### ACKNOWLEDGMENT

This work was supported by the JSPS KAKENHI Grant Number JP18K18022.

#### REFERENCES

- [1] H. Kawase, Y. Mori, H. Hasegawa, and K.-c. Sato, "Dynamic router performance control utilizing support vector machines for energy consumption reduction," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 860–870, 2016.
- [2] T. Song, Z. Jiang, Y. Wei, X. Shi, X. Ma, O. Ormond, M. Collier, and X. Wang, "Traffic aware energy efficient router: Architecture, prototype and algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3814–3827, 2016.
- [3] Y. Li, L. Zhu, S. K. Bose, and G. Shen, "Energy-saving in ip over wdm networks by putting protection router cards to sleep," *Journal of Lightwave Technology*, vol. 36, no. 14, pp. 3003–3017, 2018.
- [4] P. Liu, S. R. Chaudhry, X. Wang, and M. Collier, "Progres: A programmable green router with controlled service rate," *IEEE Access*, vol. 7, pp. 143 792–143 804, 2019.
- [5] K. Sklower, "A tree-based packet routing table for berkeley unix," in *Proc. of USENIX Winter Conference*, 1991, pp. 93–104.
- [6] S. Nilsson and G. Karlsson, "Ip-address lookup using lc-tries," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 6, pp. 1083–1092, 1999.
- [7] A. J. McAuley and P. Francis, "Fast routing table lookup using cams," in *Proc. of the 12th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '93)*, vol. 3, 1993, pp. 1382–1391 vol.3.
- [8] F. Zane, G. Narlikar, and A. Basu, "Coolcams: Power-efficient tcams for forwarding engines," in *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 1. IEEE, 2003, pp. 42–52.
- [9] W. Li, X. Li, and H. Li, "Meet-ip: Memory and energy efficient tcam-based ip lookup," in *Proc. of the 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–8.
- [10] E. Norige, A. X. Liu, and E. Torng, "A ternary unification framework for optimizing tcam-based packet classification systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 657–670, 2018.
- [11] G. C. Sankaran and K. M. Sivalingam, "Design and analysis of fast ip address-lookup schemes based on cooperation among routers," in *Proc. of the 20th International Conference on COMMUNICATION SYSTEMS & NETWORKS (COMSNETS)*, 2020, pp. 330–339.
- [12] D. C. Feldmeier, "Improving gateway performance with a routing-table cache," in *Proc. of the 7th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '88)*, 1988, pp. 298–307.
- [13] R. Jain, "Characteristics of destination address locality in computer networks: A comparison of caching schemes," *Computer networks and ISDN systems*, vol. 18, no. 4, pp. 243–254, 1990.
- [14] X. Chen, "Effect of caching on routing-table lookup in multimedia environment," in *Proc. of the 10th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '91)*, vol. 3, 1991, pp. 1228–1236.
- [15] S. Ravinder, M. A. Nascimento, and M. H. MacGregor, "Two-level cache architecture to reduce memory accesses for ip lookups," in *Proc. of the 23rd International Teletraffic Congress (ITC)*, 2011, pp. 278–285.
- [16] Y. Tung and H. Che, "A flow caching mechanism for fast packet forwarding," *Computer Communications*, vol. 25, no. 14, pp. 1257–1262, 2002.
- [17] S. Gamage and A. Pasqual, "High performance parallel packet classification architecture with popular rule caching," in *Proc. of the 18th IEEE International Conference on Networks (ICON)*, 2012, pp. 52–57.
- [18] H. Yamaki, H. Nishi, S. Miwa, and H. Honda, "Data prediction for response flows in packet processing cache," in *Proc. of the 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018, pp. 1–6.
- [19] K. Tanaka, H. Yamaki, S. Miwa, and H. Honda, "Multi-level packet processing caches," in *Proc. of the 2019 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, 2019, pp. 1–3.
- [20] M. Okuno, S. Nishimura, S. Ishida, and H. Nishi, "Cache-based network processor architecture: Evaluation with real network traffic," *IEICE trans. on electronics*, vol. 89, no. 11, pp. 1620–1628, 2006.
- [21] M. Okuno and H. Nishi, "Network-processor acceleration-architecture using header-learning cache and cache-miss handler," in *Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004)*, 2004, pp. 108–113.
- [22] H. Yamaki, "Efficient cache architecture for packet processing in internet routers," in *Proc. of the 2020 Future of Information and Communication Conference (FICC 2020)*, vol. 1, 2020, pp. 338–352.
- [23] H. Yamaki, "Flow characteristic-aware cache replacement policy for packet processing cache," in *Advances in Information and Communication Networks*, vol. 886, 2019, pp. 258–273.
- [24] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, "Cacti 7: New tools for interconnect exploration in innovative off-chip memories," *ACM Trans. Archit. Code Optim.*, vol. 14, no. 2, pp. 1–25, 2017.

- [25] T. Venkata Mahendra, S. Mishra, and A. Dandapat, "Self-controlled high-performance precharge-free content-addressable memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 8, pp. 2388–2392, 2017.
- [26] S. Mishra, T. V. Mahendra, and A. Dandapat, "A 9-t 833-mhz 1.72-fj/bit/search quasi-static ternary fully associative cache tag with selective matchline evaluation for wire speed applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 1910–1920, 2016.
- [27] WIDE MAWI WorkingGroup, "Mawi working group traffic archive," <http://mawi.wide.ad.jp/mawi/>, (Accessed on 10/01/2019).