A Dynamic Longest Prefix Matching Content Addressable Memory for IP Routing

Satendra Kumar Maurya and Lawrence T. Clark, *Senior Member, IEEE*

***Abstract*—An internet protocol (IP) router determines the next hop for a packet by finding the longest prefix match. This lookup often occurs in ternary content addressable memory (TCAM), which allows bit masking of the IP address. In this paper, an internet protocol content addressable memory (IPCAM) circuit that directly determines the longest prefix match to the stored address is described. The proposed IPCAM produces an encoded prefix match length that is limited by the prefix mask. Entries need not be sorted in order. One of the proposed IPCAM entries replaces on average 22 TCAM entries. Consequently the longest prefix matching CAM is less than 1/10 the size of the equivalent TCAM and dissipates 93.5% less dynamic power. The encoded outputs drive a priority encoder to determine the longest prefix match in the IPCAM arrays. A priority encoder circuit architecture appropriate to the unsorted IPCAM entries is also presented.**

***Index Terms*—Associative memories, internet protocol (IP) routing, longest prefix match, ternary content addressable memory (TCAM), priority encoder.**

I. Introduction

THE internet protocol (IP) has the task of delivering distinguished protocol datagrams (packets) from the source host to the destination host, based solely on their destination addresses. The IP has worked extremely well, allowing exponential growth of the internet. Initially, IP addresses were divided into the five categories, known as classes. To expand the usable IP address space, classless inter-domain routing (CIDR) was implemented [1]. CIDR allocates IP addresses in variable-sized blocks without regard to the previously used classes. CIDR was initially implemented for IPv4 where the address length is 32-bits. With continued internet growth, this address range is being exhausted. Consequently, IPv6 with 128-bit addressing is being introduced.

A. IP Router Functions

To forward packets toward their final destination the router has to perform address lookup, buffering, scheduling, and finally, send the packet to the next hop address through the appropriate router port. The address lookup, being associative, is a key processing bottleneck. Packets are routed on a next-hop basis, i.e., the router sends an incoming packet to the next hop only—the packet reaches its final destination in multiple hops. Each router has a database, in the form of a routing table containing prefixes of varying length and for each, their corresponding next hop port (NHP).

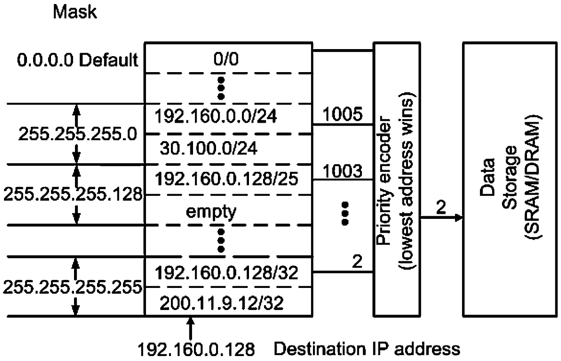

Fig. 1. Basic IP router logical structure. The values after the slashes indicate the mask values. Final hops are in external memory as shown.

B. Longest Prefix Matching

Routing based on longest prefix matching essentially routes the packet to a location as close as possible to the destination. The destination address of an incoming packet is compared with all of the current prefixes in the routing table to determine the next hop associated with the longest matching prefix. If no prefixes match the destination IP address, the packet is sent to a default port. The length of the valid part of addresses can vary up to 32 bits in IPv4, and up to 128 bits in IPv6. Mask bits determine the valid lengths of the address, i.e., address bits for which mask bits are "1" are valid and the rest of the address is ignored (see Fig. 1) working from MSB towards LSB.

Fig. 1 shows a conventional routing table implementation, where the addresses are grouped and ordered by mask size. The mask associated with each address is also shown. For instance, when a destination IP address of 192.160.0.128 is compared with the prefixes in the table, it matches with the address stored at locations 2, 1003, and 1005 but the priority encoder (PE) selects the location 2 since it has the longest prefix match. This pointer is used to read the NHP information stored in SRAM or DRAM. Ordering the entries makes selecting the longest prefix match straightforward—these operations resemble leading zeros detection, since the bottommost match (logic 1) in the table is selected.

C. Paper Organization

This paper is organized as follows. Section I has described the high level router and CIDR functionality. Section II briefly surveys previous IP lookup approaches. Section III describes a specialized IPCAM circuit for finding the next hop, first described in [2] but improved here to further reduce the energy

Manuscript received August 04, 2009; revised November 16, 2009. First published March 18, 2010; current version published May 25, 2011.

The authors are with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: smaurya@asu.edu; lawrence.clark@asu.edu).

Digital Object Identifier 10.1109/TVLSI.2010.2042826

per search. Section IV describes the priority encoder circuits required to allow determination of the NHP by determining the best (longest) of the matching prefixes. A standard TCAM implementation on the same CMOS 65-nm fabrication process is used for comparison. Extensions to IPv6 are discussed in Section V. Section VI concludes this paper.

II. Previous Work

Latency, IP address storage density, power dissipation (energy per search) and finally, determination of the global best match are the key considerations for routers. This section reviews the prior work, emphasizing the impact on these key aspects.

A. Software IP Lookup

Software approaches have the advantage of programmability, but the associative lookup requires multiple clock cycles. A tree-based data structure can be used for IP address storage and lookup [3]. For IPv4, the longest prefix length may be 32 bits so an IP lookup requires up to 32 memory accesses. To decrease the memory accesses required, a complete binary tree expansion has been proposed [4] but this requires an array with 2^{32} entries. A forwarding table scheme reduces the memory storage size and accesses, but is also large [5]. In general, any software approach on standard microprocessors must comprehend issues such as the impact of cache misses, the number and latency of memory accesses, and multiple processor clock cycles for search execution.

B. Hardware IP Lookup

IP routing hardware mostly concentrates on matching the destination address with the addresses in the routing table, which while only part of the IP lookup problem, is, as mentioned, the bottleneck. Wade *et al.* proposed an addressable search engine [6] using a TCAM structure for a database accelerator chip and a modified ripple chain priority encoder [7]. Chuang *et al.* also proposed using CAM structures [8]. Pei *et al.* implemented a high radix tree in silicon for exact matching, using a CAM-based forwarding table [9]. Degermark *et al.* used SRAM and improved the performance by converting the forwarding table radix tree to a complete tree by filling the empty branches, requiring at most four memory accesses [10]. Gupta *et al.* proposed a two memory access, two-level indirect lookup scheme. Adding a length field to the first (segment) table that maintains the length of the second (offset) table allows a variable offset and thus more efficient memory utilization [11].

C. CAM-Based IP Lookup

CAM provides a one clock matching solution [12], [13]. Fig. 1 is the top level architectural view of a TCAM based router. All valid combinations of w-bit IP prefixes may require as many as $2^{w+1} - 1$ entries, i.e., one for the null prefix, covering all entries, plus as many 1 through up to 32-bit prefixes as needed. Hayashi *et al.* described a CAM-based design with CAM blocks that have a single global mask register, saving on mask storage [14]. Akhbarizadeh *et al.* encoded the mask bit in the address for each 8-bit block using nine SRAM cells and a more complicated match line structure [15]. This circuit still requires masking each individual entry for one to w bits of width to find the longest match, but reduces mask bit storage.

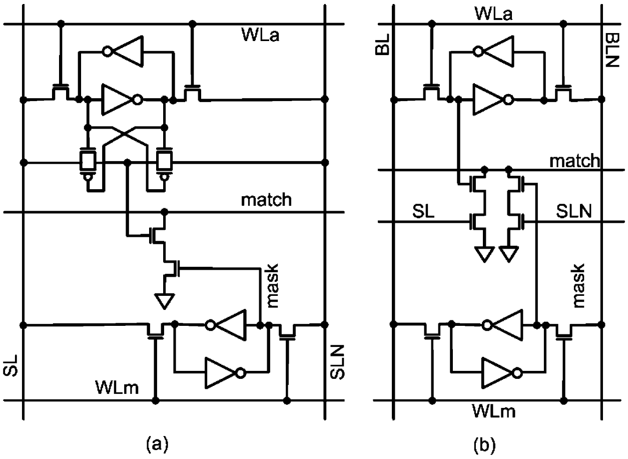

Fig. 2. Reference TCAM cells.

The conventional dynamic NOR match line discharges on a mismatch, causing high power dissipation due to the high match line activity factor, since most entries don't match the incoming address. Series transistor connected (NAND) match lines have been proposed to reduce power [16], [17] as well as combinations of NAND and NOR [18]. Large transistor stacks invite charge sharing issues, which can be addressed by building the match line from a hierarchy of short stacks [19], or by precharging intermediate nodes [20], [21]. Modern high density TCAM arrays can dissipate up to 15 W per chip [22], a multiplicity of which, are required in a high end router.

Kasai *et al.* achieved low power using charge injection match detect circuits [23]. Distributed architectures have also been proposed to reduce the search power [24]. These include arranging the forwarding table as disjoint sets [25] and segmenting the forwarding table based on the prefix length to allow incremental updates [26]. In these architectures some sets can be exhausted while others have empty space. Kaxiras *et al.* proposed a memory architecture similar to set-associative caches [27]. A drawback of this approach is routing table inflation, as routing table prefix lengths are restricted, using a controlled prefix expansion technique.

D. Reference TCAM

To provide meaningful power, density and speed comparisons, a reference TCAM array implemented in the same bulk CMOS 65-nm process technology is used in this work.[1] The cell design is shown in Fig. 2(a). Although other cell designs are denser [see Fig. 2(b)], the design in Fig. 2(a) has the least match line capacitance and thereby, lower search power dissipation [12], [19]. It consists of two SRAM cells storing the address and mask bits, respectively. 32 cells, combined with a precharge, keeper and latch block comprise one row in an array for address comparison. The basic TCAM block has up to 31 entries for a 32-bit IPv4 address, although on average the number of entries is less as described in Section III-D.

[1] [Online]. Available: http://www-03.ibm.com/technology/foundry

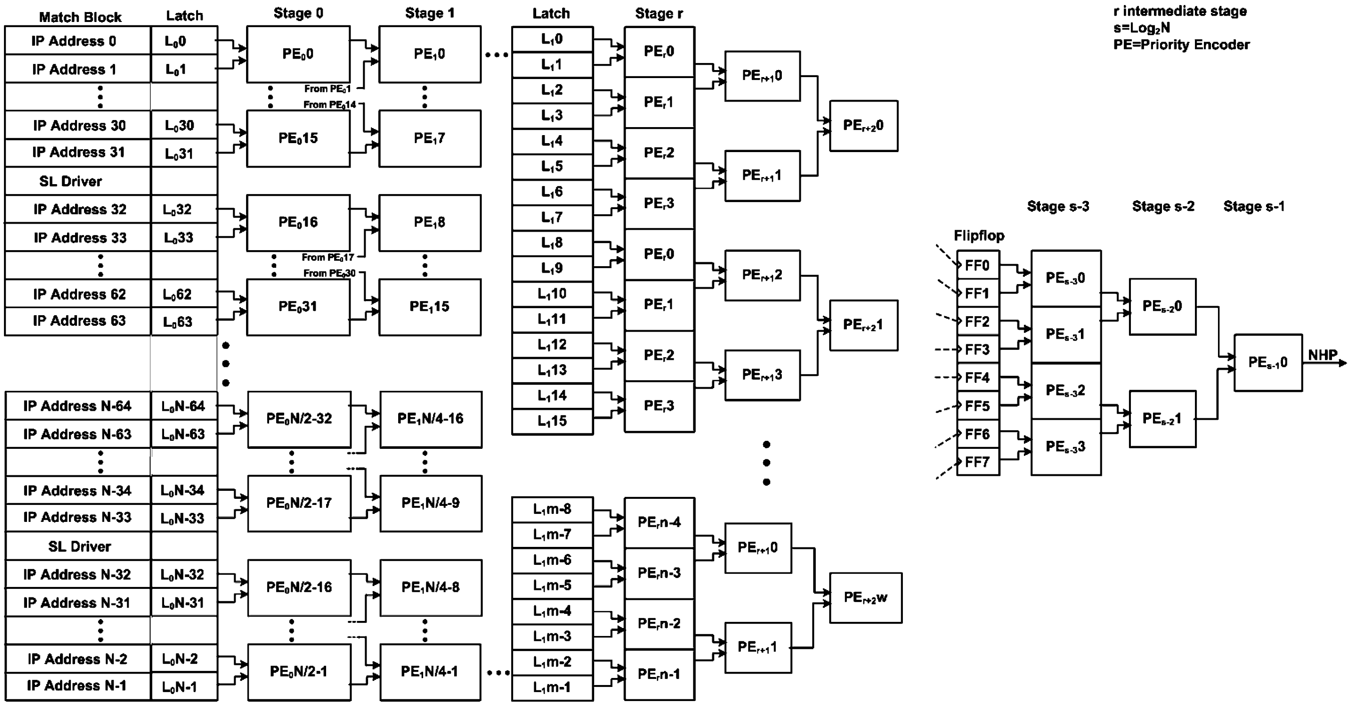

Fig. 3. Architectural details of proposed routing table circuit. The match block is composed of IPCAM circuits, followed by a transparent clock high latch and the priority encoder. Both the match block and the priority block arrangement are shown. The next hop address pointer NHP, corresponding to the location of the best match, is output.

E. Priority Encoding

Determining the number of, and which, entries in the forwarding table matching the incoming destination IP address determines the potential next hops. The next hop corresponding to the longest prefix match, which represents the optimal choice, has to be determined by a priority encoder (PE). In the conventional TCAM finding the longest match is equivalent to finding the match closest to the bottom of the lookup table, similar to leading zeros detection. For this function, a multilevel look-ahead design using domino logic has been proposed [28]. When cascading from one stage to the next, signals must be domino compatible (monotonic) and these circuits impose large clock loading. A static, modular, scalable approach [29] may also be suitable, which compares the match information hierarchically.

III. IPCAM-Based Next Hop Table

A. Overall Circuit Architecture

The proposed next hop table architecture is shown in Fig. 3. The forwarding table consists of N address entries, stored in the IPCAM, which is a dynamic circuit that performs the search in the high clock phase and precharges in the low clock phase [2]. Using the input address, each entry in the proposed IPCAM match block directly computes the longest matching contiguous bits from a single stored address and mask word. Consequently, the number of table entries is reduced by up to 31 × over the TCAM approach. The match block operates on all the N entries in parallel. Each entry determines the number of MSB bits of the stored address that match the input destination address. The result is latched at the falling clock edge, and passed to the PE, allowing the IPCAM circuit to precharge.

The IPCAM entries need not be, and in fact cannot be sorted in match length order, since any entry can match from zero up to its mask length bits. Consequently, the conventional PE cannot be used. The PE proposed here is inspired by, but is different from, that in [29]. It essentially sorts the match lengths output by the IPCAM circuit, forwarding the best value at each stage. Each PE sorting circuit operates on two sets of inputs at a time and generates as its output the best match count and the associated best matching address. Thus, a binary tree of the 2:1 PE sorting circuits is used to compute the overall longest prefix match as shown in the Fig. 3. At the end, the address of the best match (the NHP) is output to determine (the address of) the corresponding next hop address.

The number of PE sorting stages required for N addresses is $\mathrm{Log}_2 N$. The total number of PE sorting circuits required is $N - 1$. Since the PE operation takes much longer than the IPCAM lookup, the PE is pipelined. The clock period depends on the IPCAM match block delay (D_m). Thus, the PE path uses 2 D_m/D_p sorting circuit pipeline stages, delivering one match length and address per clock cycle, where D_p is the delay of each 2:1 PE sorting circuit. The latch after the match block allows time borrowing, i.e., the PE operation can begin in the first clock phase.

B. IPCAM Match Circuit

The IPCAM match block circuit is shown in Fig. 4. Each IPCAM entry contains a single address, with seven segmented match lines labeled M(A-D)0–6 and four group match lines labeled (A-D)match [2]. The circuit is divided into groups of eight bits (labeled group A-D) to limit the capacitive loading, i.e.,

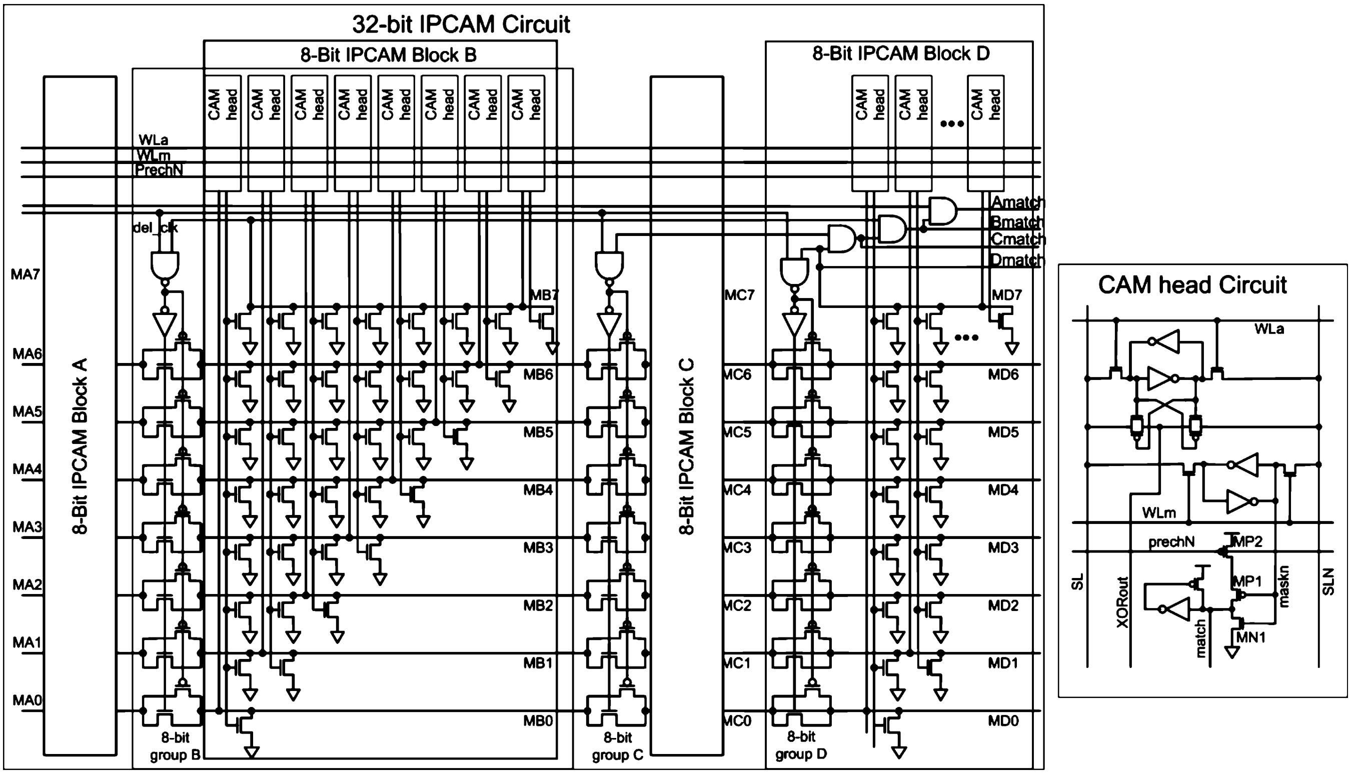

Fig. 4. Proposed IPCAM row. Each row encodes the longest matching prefix in signals MD6-MD0 and (D-A)match. Each group A-D contains 8 bits. Search/bit lines are not shown for clarity, but are routed vertically (MSB's are on the right). The CAM head circuit is shown at the right.

fan-out, of each circuit, and to allow a shorter match length encoding. Each CAM head circuit drives from one to eight match line pull down transistors.

In operation, one of the clock ANDed differential search lines for each of the 32 columns is asserted high in the first clock phase, starting a match operation. The column-wise XOR network in the CAM head cell determines if the stored address bit matches the incoming address bit for that column. If it does not, signal XORout (the vertical signal produced by the CAM head circuit) is asserted high. Each match line connected to that CAM head cell is then discharged. The groups of eight columns have a triangular configuration, i.e., the leftmost column can discharge any of the eight match lines, but the rightmost can discharge only the topmost match line, e.g., MD7. The critical timing delay path is thus through the group A column driving eight pull down transistors, the match line with eight pull down transistors (e.g., MD7), the NAND gate, inverter and finally MA0-6 values through the B through D 8-bit group match lines. There is a race between the $\mathrm{MB}\langle 6:0\rangle, \mathrm{MC}\langle 6:0\rangle$ and $\mathrm{MD}\langle 6:0\rangle$ in the CAM evaluation and best match propagation modes. The delayed clock signal del_clk must arrive after the 8-bit groups have evaluated, to open the pass gates. This is necessary since masked match lines are always driven low and these bits would interfere with the pre-charging in subsequent groups. The full group match lines M(A-D)7 are not multiplexed in the same manner as the others; all of them must be output.

When an entire group matches, i.e., all 8 bits in the group match the incoming address, that group signals out on one of the signals (A-D)match that this has occurred by asserting (A-D)match. For instance, if the 8 MSB's match, node MD7 stays high, it directly asserts node Dmatch (node MD7's alias). If the next 8-bit group matches, then Cmatch is asserted to indicate a 16-bit match. The AND gates ensure that proper codes are output. The match lines are reused so allow transfer of the subsequent (the group to the left) 8 bit group's match information through the same match lines. This limits the metal usage as the cell block is metal limited. If 8-bit groups C and D fully match, but there is a mismatch at the 5th bit in group B, then the CMOS pass gates for groups C and D are opened. The output signals MD0-6 indicate the state of the group B match lines MB0-6. Assuming 4 bits match in group B, the outputs are $\mathrm{Amatch} = \mathrm{Bmatch} = 0, \mathrm{Cmatch} = \mathrm{Dmatch} = 1, \mathrm{MD0} - 3 = 1$ and $\mathrm{MD4} - 6 = 0$. As obvious from an examination of the circuit, the (A-D)match and MD0-6 lines output thermometer codes.

The CAM head cells are written and read by placing the data to be stored on the combination search/bit lines SL and SLN and asserting the WLa word line to write the address storage or the WLm word line to write the mask storage. This aspect of the circuit is completely conventional. The IPCAM match lines are made pseudo-static by the pMOS keeper transistors on each of them (see Fig. 4). The circuit in [2] used a long channel keeper pMOS transistor to ensure domino node write ability. Here, we reduce the gate overdrive of the pMOS transistor by one PMOS transistor threshold voltage. Referring to Fig. 4, a pMOS pull down transistor limits the keeper transistor V_{GS}. This configuration reduces the keeper transistor capacitance and thus power dissipation by 6.6% on a match line discharge, when compared to that in [2].

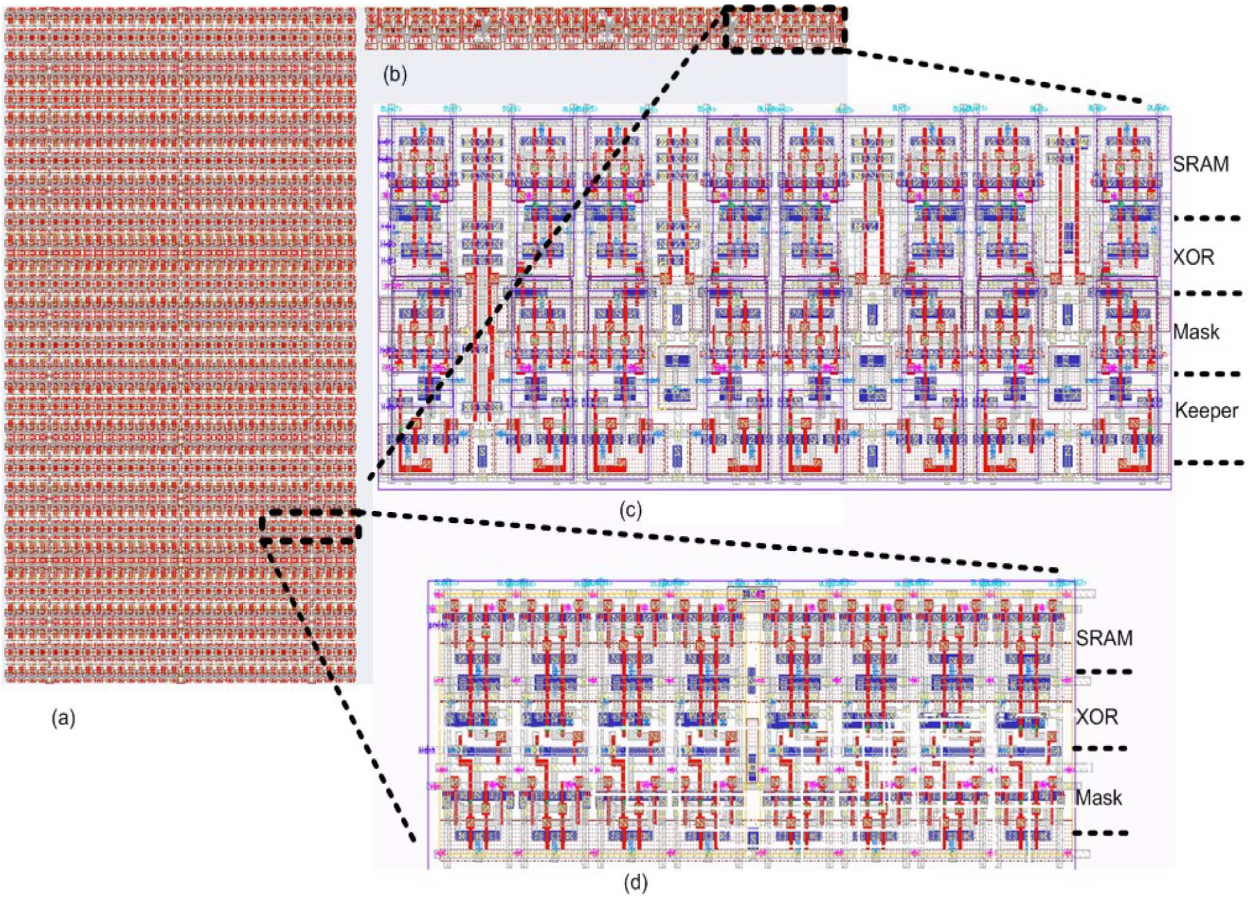

Fig. 5. IPCAM row layout and area compared to TCAM cells. 22 entry TCAM array (a) and the equivalent storage to match prefixes up to 22-bits using IPCAM (A 32-bit IPCAM) (b) shows the area improvement of the proposed approach. The individual circuit portions are also shown for eight IPCAM bits (c) and eight TCAM bits (d) to show the circuit details. The IPCAM circuits are larger but they replace on average 22 TCAM entries, since a TCAM entry must exist for each prefix length. (a) 22 × 32 bits of TCAM cells, (b) 32 bits of IPCAM, (c) 8 bits of IPCAM, (d) 8 bits of the TCAM cells.

C. Masking

Referring to Fig. 4, prefixes are stored with the MSB to the right and the LSB to the left. The mask bits are set from left to right. For instance, if the prefix length is 24 bits, then group A is left out of the prefix search for that entry. The longest prefix that can match is 24 bits, so these match lines are permanently discharged by the mask bits in the CAM head cell. These match lines are never pre-charged, since the pMOS transistor MP1 in the pMOS stack composed of transistors MP1-2 (see Fig. 4 right) disables that operation. To avoid failure due to leakage or noise, the match line is held low by the mask bit controlled transistor MN1. Thus, only search line power is dissipated in masked off bits in each table entry. Returning to the example where all bits in group A are masked, the maximum output code is then $\mathrm{Amatch} = 0$, $(\mathrm{B-D})\mathrm{match} = 1$ and $\mathrm{MB0-6} = 0$ indicating a 24 bit match.

The search line drivers are placed in the bank center to drive 32 addresses differentially to entries both above and below them. Signals SL and SLN are driven low during the precharge phase of the clock allowing match lines to precharge. For IPv4, 32 search lines are needed for each address. Hence a total of $32 \times (N/64)$ search line drivers are required.

D. IPCAM Area

While the IPCAM design matches up to 32 bits, the actual power and area savings is less than that found by calculating based on one entry in the IPCAM and 32 entries in an equivalent TCAM. Whereas a TCAM row is required for each match length, only a handful of addresses are 32-bits long, since this fully specifies a destination.

The border gateway protocol (BGP) routing tables contain nearly 220 K entries.[2] The average prefix length is 22, with 24-bit prefixes comprising 53% of the entries. Consequently, for power and area comparisons between our IPCAM and equivalent TCAM circuits we use the average BGP table prefix length of 22. The following analysis assumes the CAM search line drivers drive 64 rows of either TCAM or IPCAM cells, which was used in all simulations and layouts. We treat the search line driver power separately, since considerably more are needed for the equivalent capacity TCAM array.

The IPCAM and PE designs, as well as a representative TCAM array, have been implemented in a foundry bulk CMOS 65-nm technology.[1] This allows simulations using extracted values from the layout, properly accounting for wire loading and resistance–capacitance (*RC*) effects on delay. Fig. 5(a) shows the layout of 22×32 TCAM cells to the same scale as one 32-bit IPCAM entry shown in Fig. 5(b). One 32-bit IPCAM entry replaces, depending on the mask settings, 22 entries on average as mentioned above. The area improvement between the 22 entry TCAM array and the single IPCAM entry is clearly evident. Layout details of the 8-bit IPCAM

[2] [Online]. Available: http://bgp.potaroo.net

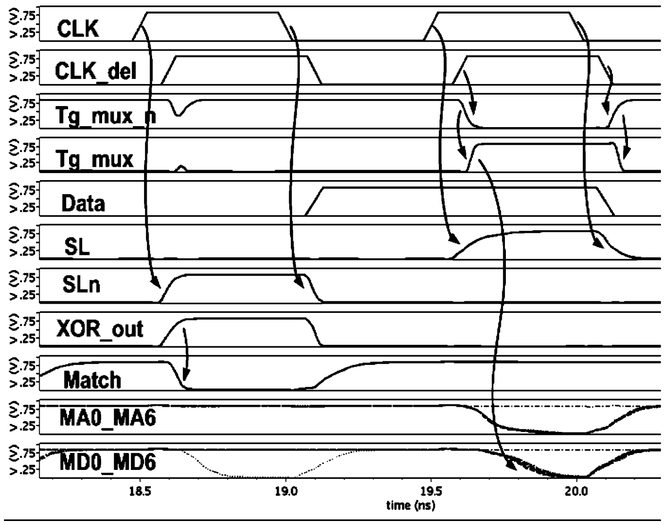

Fig. 6. Simulated IPCAM operation.

TABLE I
POWER AND DELAY COMPARISON

Architecture	Power (μW)	Mean energy (fJ/bit/search)	Delay (ps)
IP-CAM (1 entries)	72.31	2.26	385
TCAM (22 entries)	792	1.13	380
Normalized TCAM	792	24.8	380

TABLE II
IPCAM AND TCAM POWER

Architecture	Power (μW)
1-Search Line Driver	19.6
1 Entry IP-CAM (up to 32 bit match)	83.1
22-Entries TCAM (up to 22 bit match)	1040
64-Address Match using IPCAM	5945.6
64-Address Match using TCAM	80351.4

slice and eight TCAM cells are also shown in Fig. 5(c) and (d), respectively.

Each TCAM cell requires 18 transistors [see Fig. 2(a)]. Implementing a 22-bit (maximum match length) address the average number of entries require $12\,672 (= 22 \times 32 \times 18)$ transistors for the TCAM array. The same prefix match capability in the proposed architecture uses 1532 transistors. The array savings is thus 88%. Compared to the densest TCAM circuit [see Fig. 2(b)] design for comparison, this advantage is 86%. Each TCAM cell [see Fig. 2(a)] in the target process is 1.31 by 3.46 μm. 32 × 22 cells thus occupies 3199 μm^2. Each 32 bit IPCAM entry is 67.15 by 4.72 μm occupying 317 μm^2. Consequently, the proposed IPCAM density is approximately 10 × better.

E. Speed and Power and Area

The search/bit line drivers drive 64 rows of TCAM or 64 rows of IPCAM. Thus, approximately 22 × as many search/bit line (SL and SLN in Fig. 4) drivers are required for the TCAM as for the equivalent IPCAM longest prefix match search capacity. In the IPCAM, the worst-case search lines are more heavily loaded, driving eight pull down transistors for each entry. This makes the IPCAM search lines slightly slower with the same drive strength. The TCAM match line nMOS pull down transistors are sized to provide the same discharging current.

The TCAM and IPCAM power dissipation are determined by circuit simulation including parasitic capacitances and wire resistances extracted from the layout using Calibre PEX. We separated the search/match line driver power dissipation from that of the CAM arrays since the IPCAM requires far fewer of them.

Fig. 6 shows the simulated IPCAM operation. Limiting the design to 8-bit groups limits fan out on the CAM head XORout signals and also limits the group match line delay. The delay of the 32-bit IPCAM circuit is 385 ps from the clock assertion to the last match line signal out on MD6, where only one match line pull down nMOS transistor discharges the match line. The dynamic match results must be latched to hold their values in the subsequent clock phase. The TCAM design has similar delay in the worst-case, with one nMOS pull down transistor active, with a clock to match line discharge delay of 380 ps. Since the match occupies the first clock phase and precharge the second, both the TCAM and IPCAM can operate at better than 1 GHz clock frequencies in the target process.

Table I compares the power and delay for the two circuit architectures. Each IPCAM entry is equivalent to $22 \times 32 (= 704)$ entries of TCAM for similar match output. Hence 64 IPCAM entries in one sub-array is equivalent (on average) to 1408 32-bit TCAM entries. The TCAM (requiring 22 32-bit entries) has 704 bits of storage compared to the equivalent single 32-bits plus mask IPCAM entry. The normalized TCAM energy/bit/search accounts for the TCAM requiring 22 entries on average per IPCAM entry. When the TCAM energy per bit/search is normalized to be the same as the IPCAM, i.e., the address storage, rather than the larger number of bits required by the TCAM, the IPCAM circuit is shown to be about 10 × better. The simulations assume that the match lines miss and are discharged, since that is the common case in a large CAM, e.g., 64 k entries. Table II shows the power dissipation for 64 address entries, including that of the search line drivers. Hence for IPCAM 64 × 32 search lines are required. However, for TCAM 64 × 32 × 22 search lines are required. We assume one search line driver for every 64 entries, so the TCAM simulations include 22 more of those. For a specific address, many match lines will not discharge, but statistically, this number is insignificant—in the simulations we assume that all discharge.

Table III shows example match lengths and their output code values, as well as the power, energy per bit/search, and delay in the IPCAM. This simulation uses a 32 × 64 entry array and the search/bit line drivers. All columns participate in the match operation. This makes the simulation worst-case (skewed to disfavor the proposed IPCAM) since masking reduces this and on average 10 bits will be masked. Power dissipation depends on which 8-bit set is selected for the output. The worst case delay is for a 25-bit match length. In this case the signals MA0-MA6 have to propagate through the following three 7-bit groups, which all match. The propagation delay from driving

TABLE III
IPCAM SIMULATED POWER AND DELAY FOR A SAMPLING OF MATCH LENGTHS AND THEIR OUTPUT CODES

Match Length	Power (mW)	Energy (fJ/bit/search)	Delay (ps)	Output Code
32	4.669	2.27	66.42	1111_1111111
31	5.010	2.45	191.9	0111_1111111
30	5.204	2.54	358.7	0111_0111111
29	5.200	2.54	307.6	0111_0011111
28	5.238	2.56	364.9	0111_0001111
27	5.192	2.54	309.1	0111_0000111
26	5.154	2.52	310.4	0111_0000011
25	5.294	2.58	381.1	0111_0000001
22	5.168	2.52	254.2	0011_0111111
19	5.085	2.48	277.6	0011_0000111
15	5.163	2.52	241.7	0001_1111111
8	5.082	2.48	248.4	0001_0000000
7	4.760	2.32	196.7	0000_1111111
2	5.101	2.49	243.8	0000_0000011
1	5.154	2.52	244.2	0000_0000001
0	5.118	2.50	245.8	0000_0000000

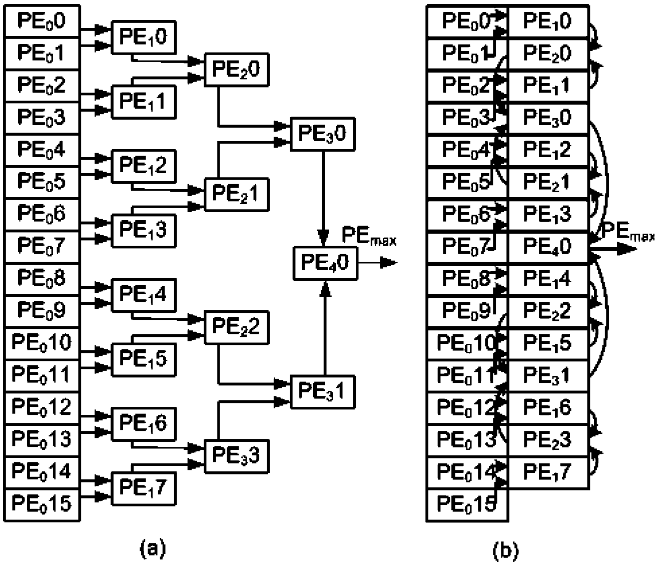

Fig. 7. (a) PE with 5-stages of PE sorting circuits and (b) the individual block arrangement and interconnections.

the lower seven bits from the first IPCAM 8-bit group through the others, dominates the delay (see Fig. 6).

IV. PRIORITY ENCODER

A. Priority Encoder Architecture

While each entry generates the longest match between it and the input IP address, the best match, as well as its location, which implicitly points to the next hop, must be determined. The outputs generated by the IPCAM consist of two thermometric codes. Four bits are outputs A-D and the other seven bits are signals AD6-AD0.

As mentioned in Section III-A, the PE is composed of a binary tree of PE compare and forward sorting circuits, each comparing input vectors $P\langle 10:0\rangle$ and $Q\langle 10:0\rangle$, comprised of the IPCAM match circuit outputs A-D concatenated with outputs AD6-AD0 or the same vectors from a previous stage. The maximum of P and Q is dominated by the upper four bits, which determine the number of 8-bit groups matching. Thus, if the $P\langle 10:7\rangle$ is greater than $Q\langle 10:7\rangle$ then P and its associated IPCAM entry address is assigned to the sorting circuit output R. Otherwise, Q and its associated IPCAM address is output. However when these MSB bits are equal for P and Q then R is assigned based on the best match length as described by the lower order bits $P\langle 6:0\rangle$ and $Q\langle 6:0\rangle$.

Fig. 7 shows the basic organization, which, due to there being 1/2 as many sorting circuits at each subsequent level, can be laid out in two columns. Latches are required to hold the outputs of the dynamic IPCAM circuits during the precharge clock phase. The priority encoding begins in the first clock phase, as soon as the IPCAM match block outputs are valid. This time-borrowing allows 6 PE stages in the clock cycle after the match block, although another transparent latch is required within that 6 PE stage unit. The rest of the pipeline stages use master-slave flip-flops. The height of each PE compare block is equal to the height of two IPCAM match block entries. In order to minimize the wire length, the 5-stage PE is placed in the middle. For a 64 k entry IPCAM, the total chip area is approximately 5 × 5 mm, so the maximum wire length is about 2.5 mm. Simulations using the foundry supplied interconnect *RC* models were used to optimize the number of inverting repeaters. They are placed every 500 μm.

B. PE Comparison Circuit

Several circuits that compare the input vectors and output the greater of the two have been proposed. Dmitry *et al.* proposes comparators that dissipate power on a match as compared to the traditional domino circuit which dissipates dynamic power on mismatch [30]. Wang *et al.* proposed high fan-in dynamic CMOS comparators [31]. We opted for a simple static comparison circuit, to avoid high clock power and because it allows easier PE pipelining. Additionally, static logic affords a significant reduction in power dissipation. Fig. 8 shows the PE sorting circuit, composed of a comparator and forwarding multiplexers. Vectors $P\langle 10:0\rangle$ and $Q\langle 10:0\rangle$ are the two 11-bit outputs from either the IPCAM or previous stages. The greater from any set of thermometric codes can be obtained as

$$(P > Q) = \sum_{k=1}^{n} (P_k Q'_k) \tag{1}$$

where n is the number of bits in the thermometric code. The thermometer encoding greatly simplifies the comparisons. Basically, logically ANDing the complement of one vector Q with the other P and then logically ORing the resulting bit vector determines if P is greater.

Referring to Fig. 8 signals pgrtr and plss correspond to the first group of the thermometric code (10 bits to 7). Signal plss is generated using the same circuit as for the pgrtr signal but with the opposite true and complement input vectors. Signal plss_lsb corresponds to the next group of thermometric codes (6 bits to 0). If those match, the choice is arbitrary. The signals pgrtr plss and plss_lsb are defined with respect to signal $\mathrm{P}\langle 10:0\rangle$, i.e.,

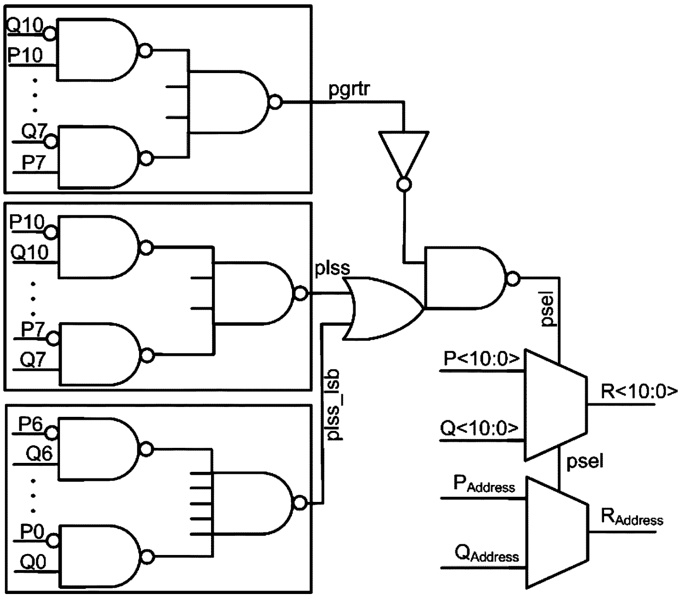

Fig. 8. PE sorting circuit schematic. The comparators determine the best matching thermometer code encoded best match length. The corresponding IPCAM entry address is output on $R\langle 10:0\rangle$ and $R_{\rm address}$, respectively.

TABLE IV
TRUTH TABLE FOR SIGNAL PSEL

pgrtr	plss	plss_lsb	psel
0	0	0	1
0	0	1	0
0	1	X	0
1	0	X	1
1	1	X	X

signal pgrtr signifies is $P\langle 10:7\rangle$ is greater than $Q\langle 10:7\rangle$. The 7-input NAND gate required for generating the signal plss_lsb is implemented using two logic stages (inversions). Signal psel selects the longest matching prefix from the two sets of incoming match by controlling the output multiplexer. The fact that both pgrtr and plss can never be asserted high simultaneously is exploited in the psel signal generation. Table IV shows the details of the psel signal generation. "X" denotes don't care conditions, of which the simultaneous assertion of pgrtr and plss is the most important. The 2:1 PE sorting circuit requires five lightly loaded inversions to generate psel, which controls the multiplexer. The next hop address corresponding to the matches P and Q can similarly be muxed using the signal psel and passed on to the next stage.

C. Performance and Power

The PE sorting circuit delay and the power computations have been determined by circuit simulations while driving the inputs with various combinations. Simulated waveforms for the worst case timing are shown in Fig. 9. The worst case delay of 143 ps is as expected, when Q is selected for output, based on the lower thermometric code, i.e., $Q\langle 6:0\rangle$. Due to low fan-out on each gate, five stages can fit in a 1 GHz clock cycle. PE sorting circuit power dissipation of 16.66 μW at 1 GHz clock rate keeps the overall PE power dissipation low.

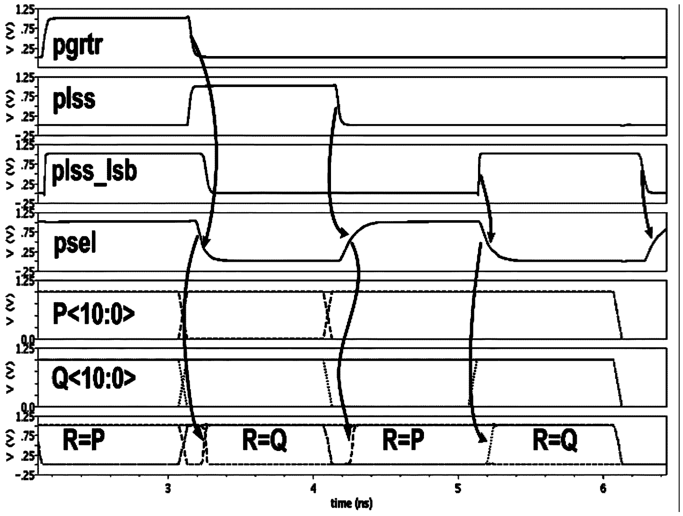

Fig. 9. Simulated static priority encoder operation and timing.

TABLE V
RESOURCE UTILIZATION FOR 64 K ADDRESS CAM

Resource	Count	Power (W)
IPCAM (entries)	65536	5.45
Search line drivers	32768	0.64
PE (PE sort circuits)	65535	1.09
Flip-flops	11627	0.03
Latches	90112	0.12
Total		**7.33**

For a 64 k entry IPCAM IC in which each search line drives 64 IPCAM entries, 32 k search line drivers are required. The PE requires 16 stages of PE compare. 1024 6-stage PE sort circuits and 33 5-stage PE sort circuits are required. A total of 11 627 flip-flops and 90 112 latches, including those for the IPCAM outputs, are required. Table V details the resource requirements and the power associated with them. The overall 64 k entry IC power dissipation is dominated by the dynamic IPCAM block.

A TCAM-based implementation requires leading zero detection to determine the longest match, i.e., the bottom most matching entry (refer to Fig. 1). In the target technology a 32-bit leading zero detector using static logic has a delay of 115 ps. This operation requires one additional clock cycle since five 32-bit stages are needed. Consequently, the TCAM latency is lower compared to our design.

V. EXTENSION TO IPV6

The proposed IPCAM architecture can be extended to IPv6. By concatenating four IPCAM blocks and grouping the outputs, an IPv6 address lookup can be realized. By operating the four blocks in parallel, the achievable clock rate is still above 1 GHz, exceeding current internet requirements. For instance, the current state of the art 10 G Ethernet supports 10 Gb/s data transmission. Assuming the minimum packet size of 64-bytes, sequential worst-case address lookups require 156 MHz operation, assuming one lookup per cycle. As per the BGP table for IPv6, the average prefix length is only 48 bits, so operating on all 128 bits in parallel wastes significant power dissipation.

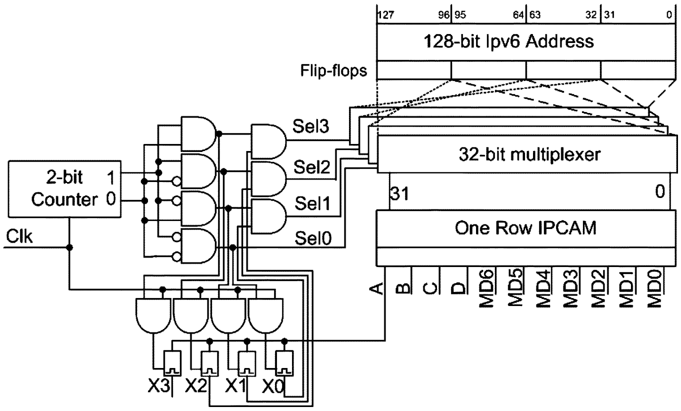

Fig. 10. Proposed IPv6 implementation.

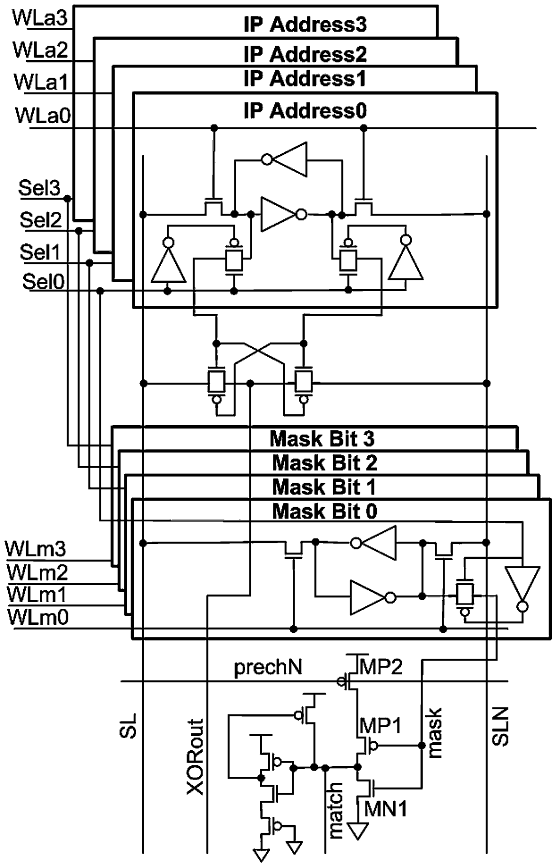

Fig. 11. CAM head circuit for IPv6 implementation.

A more efficient IPv6 implementation employs a single 32-bit IPCAM matching circuit, driven by four 32-bit data and mask memory registers. This circuit, outlined in Fig. 10, performs a matching operation spanning four consecutive clock cycles with one 32-bit comparison each cycle. A 2-bit counter controls which 32-bit block to compare based on the matching information. The CAM head block diagram for this implementation is shown in Fig. 11. The address and corresponding mask values are stored in the SRAM-based registers. During the match operation the comparison of the address proceeds from the MSB towards the LSB in 32-bit groups. When all the bits of a group match, the next is compared in the next clock phase. Otherwise, the comparison process is terminated early, eliminating the subsequent stage power dissipation. The output generated is 14-bits (X3-X0, B-D, MD6-0), comprised of three sets of thermometer codes. The lower thermometer counts the number of single bit matches, the middle counts the number of 8-bit matches and the upper thermometer codes signifies the number of 32-bit matches. This architecture saves area by reducing the compare arrays outlined in Fig. 4 by 3/4, while reducing the average power dissipation by 50%. The address lookups easily meet the required cycle times for the worst-case 64-byte packets.

VI. Conclusion

An IPCAM circuit architecture that directly calculates the number of sequential matching bits (the longest matching prefix) for 32-bit address, i.e., for IPv4, has been described in detail. By directly calculating the matching prefix length, which is output as thermometer codes on 11 signals, one 32-bit entry provides the equivalent of approximately 22 32-bit TCAM entries, based on the average prefix length in the BGP tables. While a single 32 bit IPCAM entry is about 2.2 × the size of a 32-bit TCAM entry, the 22 × advantage in the number of entries required improves the density of the proposed IPCAM design over the conventional TCAM design by 10×. This size advantage translates directly to power savings, which is also better than 90%.

Since the IPCAM cannot provide the matches ordered by length, it must be coupled to a sorting, rather than leading 1's detecting priority encoder architecture. This PE circuit, implemented as a binary tree of two-input, single-output sorting circuits, has also been described. The thermometric codes output by the IPCAM facilitate the comparisons, which do not require XOR gates. A simple static implementation allows the PE to dissipate less than 20% of the overall power for a 64 k entry IPCAM based routing table IC. 64 k entries is equivalent to approximately 1.44 M TCAM entries, assuming the average of 22-bits per entry. The IPCAM block power dissipation increases linearly with the number of entries. However, PE power dissipation increases exponentially. Estimates based on a modest 5 × 5 mm die with 64 k entries shows that larger IPCAM ICs would be practical.

Extension of the proposed IPCAM design to IPv6 has also been described. By operating one 32-bit matching block over four clock cycles, an IPv6 design can meet the current internet speed requirements with an approximately 50% increase in average match operation power dissipation.

References

[1] F. Baker, "2.2.5.2 Classless interdomain routing (CIDR)," RFC1812, Jun. 1995. [Online]. Available: http://rfc.sunsite.dk/rfc/rfc1812.html

[2] S. Maurya and L. Clark, "Low power fast and dense longest prefix match content addressable memory for IP routers," in *Proc. ISLPED*, 2009, pp. 389–394.

[3] K. Sklower, "A tree-based routing table for Berkeley unix," Univ. California, Berkeley, 1993.

[4] P. Gupta, S. Lin, and N. McKeown, "Routing lookups in hardware at memory access speeds," in *Proc. IEEE INFOCOM*, Mar. 1998, pp. 1240–1247.

[5] L. Wuu and S. Pin, "A fast IP lookup scheme for longest-matching prefix," in *Proc. IEEE Int. Conf. Comput. Netw. Mobile Comput.*, Oct. 2001, pp. 407–412.

[6] J. P. Wade and C. G. Sodini, "A ternary content addressable search engine," *IEEE J. Solid-State Circuits*, vol. 24, no. 4, pp. 1003–1013, Aug. 1989.

[7] H. Kodata, J. Miyake, Y. Nishimich, H. Kudo, and K. Kagawa, "An 8 kb content-addressable and reentrant memory," in *ISSCC Dig. Tech. Papers.*, Feb. 1985, pp. 42–43.

[8] P. T. Chuang, R. L. Yau, H. Yoshida, and M. Wang, "Content addressable memory array with priority encoder (Patent Style)," U.S. Patent 4 928 260, May 22, 1990.

[9] T. Pei and C. Zukowski, "Putting routing tables in silicon," *IEEE Netw. Mag.*, vol. 6, no. 1, pp. 42–50, Jan. 1992.

[10] M. Degermark, A. Brodnik, S. Carlsson, and S. Pink, "Small forwarding tables for fast routing lookups," *Proc. ACM SIGCOMM Comput. Commun. Rev. 27*, vol. 4, pp. 3–15, Oct. 1997.

[11] N. Huang, S. Zhao, J. Pan, and C. Su, "A fast IP routing lookup scheme for gigabit switching routers," in *Proc. IEEE INFOCOM*, Mar. 1999, vol. 3, pp. 1429–436.

[12] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[13] A. McAuley and P. Francis, "Fast routing table lookup using CAMs," in *Proc. IEEE INFOCOM*, Mar. 1993, vol. 3, pp. 1382–1391.

[14] T. Hayashi and T. Miyazaki, "High-speed table lookup engine for IPv6 longest prefix match," in *Proc. IEEE GLOBECOM*, Dec. 1999, vol. 2, pp. 1576–1581.

[15] M. Akhbarizadeh, M. Nourani, D. Vijayasarathi, and P. Balsara, "A nonredundant ternary CAM circuit for network search engines," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 3, pp. 268–278, Mar. 2006.

[16] C. Wang, J. Wang, and C. Yeh, "High-speed and low-power design techniques for TCAM macros," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 530–540, Feb. 2008.

[17] H. Li, C. Chen, J. Wang, and C. Yeh, "An AND-type match-line scheme for high-performance energy-efficient content addressable memories," *IEEE J. Solid-State Circuits*, vol. 41, no. 5, pp. 1108–1119, May 2006.

[18] A. Efthymiou, "A CAM with mixed serial-parallel comparison for use in low energy caches," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 3, pp. 325–329, Mar. 2004.

[19] V. Chaudhary and L. Clark, "Low-power high-performance NAND match line content addressable memories," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 8, pp. 895–905, Aug. 2006.

[20] F. Shafai and K. Schultz, "Fully parallel 30-MHz, 2.5 Mb CAM," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1690–1696, Nov. 1998.

[21] N. Weste and D. Harris*, CMOS VLSI Design, A Circuits and Systems Perspective*, 3rd ed. New York: Addison Wesley.

[22] R. Panigrahy and S. Sharma, "Reducing TCAM power consumption and increasing throughput," in *Proc. IEEE Symp. High Perform. Interconnects*, Aug. 2002, pp. 107–112.

[23] G. Kasai, Y. Takarabe, K. Furumi, and M. Yoneda, "200 MHz/200 MSPS 3.2 W at 1.5 V Vdd, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2003, pp. 387–390.

[24] K. Zheng, C. Hu, H. Liu, and B. Liu, "An ultra high throughput and power efficient TCAM-based IP lookup engine," in *Proc. INFOCOM Conf. IEEE Comput. Commun. Societies*, Mar. 2004, vol. 3, pp. 1984–1994.

[25] J. Kim and J. Kim*, An Efficient IP Lookup Architecture With Fast Update Using Single-Match TCAMs*. Berlin, Germany: Springer-Verlag, 2008, pp. 104–114.

[26] V. C. Ravikumar and R. N. Mahapatra, "TCAM architecture for IP lookup using prefix properties," *IEEE Micro*, vol. 24, no. 2, pp. 60–69, Mar.–Apr. 2004.

[27] S. Kaxiras and G. Keramidas, "IPStash: A power-efficient memory architecture for IP-lookup," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarch. (MICRO-36)*, 2003, pp. 361–372.

[28] J. Wang and C. Huang, "High-speed and low-power CMOS priority encoders," *IEEE J. Solid-State Circuits*, vol. 35, no. 10, pp. 1511–1514, Oct. 2000.

[29] V. G. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: Comparison with logic synthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 1, pp. 124–128, Mar. 1994.

[30] D. Ponomarev, G. Kucuk, O. Ergin, and K. Ghose, "Energy efficient comparators for superscalar datapaths," *IEEE Trans. Comput.*, vol. 53, no. 7, pp. 892–904, Jul. 2004.

[31] C. Wang, P. Lee, C. Wu, and H. Wu, "High fan-in dynamic CMOS comparators with low transistor count," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 40, no. 9, pp. 1216–1220, Sep. 2003.

Satendra Kumar Maurya received the B.Tech. degree in electronics and communication engineering from National Institute of Technology, Trichy, Tamil Nadu, India, in 2004 and is currently pursuing the M.S. degree in electrical engineering from Arizona State University, Tempe.

From 2005 to 2007, he worked as a Senior Hardware Design Engineer with Conexant Sys. Pvt. Ltd., where he performed RTL logic design. In 2004, he was the Design Engineer with Tata Elxsi Ltd., where he worked on verification of CAN modules. In 2008, he worked as an intern with Intel on the validation of interconnects fabrics for the Atom processor. His research interests include circuits and architectures for low-power and high-performance VLSI, integrated circuits and computer architecture.

Lawrence T. Clark (S'86–M'90–SM'01) received the B.S. degree in computer science from Northern Arizona University, Flagstaff, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from Arizona State University (ASU), Tempe, in 1987 and 1992, respectively.

Prior to 1992, he worked with Intel in test engineering and VLSI Technology Inc. designing PC chipsets. He rejoined Intel in 1992 and contributed to the Pentium, Itanium, and XScale microprocessor designs, receiving an Intel Achievement Award for the latter. Most recently, he was a Principal Engineer and Circuit Design Manager for the XScale microprocessors. In 2003–2004, he was an Associate Professor with the University of New Mexico. He joined ASU in August 2004. He holds 64 patents. He has authored or coauthored over 70 peer reviewed technical papers. His research interests include circuits and architectures for low power and high performance VLSI, integrated circuit radiation hardening, flexible electronics, and CAD for VLSI.

Prof. Clark has served as a Guest Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He currently serves on the editorial board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS, and the technical committees for IEEE CICC, IEEE NSREC, and ISLPED.