Vertically Partitioned SRAM-Based Ternary Content Addressable Memory

Zahid Ullah and Sanghyeon Baeg

***Abstract*—This paper proposes a novel memory architecture called VP SRAM-based TCAM (Vertically Partitioned Static Random Access Memory based-Ternary Content Addressable Memory) that emulates TCAM functionality with SRAM.VP SRAM-based TCAM dissects conventional TCAM table vertically (column-wise) into TCAM sub-tables, which are then processed to be stored in their corresponding SRAM blocks. During search operation, SRAM blocks are addressed in parallel by their corresponding sub-words of the input word and the read out rows of which are bit-wise ANDed that results in potential matching address(s) where a priority encoder selects the highest priority matching address. Search operation in VP SRAM-based TCAM involves two SRAM accesses followed by ANDing operation. Analysis shows that maximum possible number of vertical partitions reduces size of the proposed TCAM approximately by a factor of 1.3 than its traditional counterpart and offers optimized values for both area and latency of VP SRAM-based TCAM and hence, is a practically feasible alternative to traditional TCAMs.**

***Index Terms*—Memory architecture, SRAM, ternary content addressable memory (TCAM), vertical partition.**

I. Introduction

The main role of Ternary Content Addressable Memory (TCAM) is to search input data against the pre-loaded data and output the comparison result [1], which is then used to invoke a related entry from a conventional memory. A TCAM cell has a mask cell, data cell, and masking and comparison circuitry. Mask and data cells are typically implemented with SRAM [2].

TCAM is an outgrowth of RAM, which became popular in the literature for its high speed search operation. However, the speed of CAM comes at the cost of increased silicon area and power consumption [3]. CAM devices have very limited pattern capacity and also the cost of CAM devices is much higher than RAM devices of similar sizes and access time. CAM technology does not evolve as fast as the RAM does. RAM technology is driven by many applications, particularly computers and consumer electronic products, and hence cost per bit is continuously decreasing, as opposed to CAM technology, which is considered specialized and only a modest increase in bit capacity and a modest decrease in cost may be expected in future[4].

RAM is presented in a wider variety of sizes and flavours. It is more generic and widely available and enables to avoid the heavy licensing and royalty costs charged by some CAM vendors [5]. Therefore, what is desired in the literature is an alternative to TCAM that is constructed from SRAM.

This paper presents a novel memory architecture called VP SRAM-based TCAM (Vertically Partitioned Static Random Access Memory based-Ternary Content Addressable Memory) that emulates TCAM functionality with conventional SRAM memory and inherits its advantages like low cost per bit and higher memory density.

The rest of the paper is organized as follows: Section II explains previous work. Sections III and IV elaborate the architecture and phases of VP SRAM-based TCAM. Section V analyzes the proposed TCAM and section VI provides conclusion and highlights the direction of our future work.

II. Previous Work

The most relevant RAM-based TCAM architectures as per our best knowledge from the CAM literature are referenced in this section.

In US patent [4], a RAM-based CAM has been presented, specifically targeting its application to ATM communication systems but it has some serious shortcomings. Size of this scheme basically depends on the number of bits in a pattern (p). The required memory size would be 2^p bits. Size will increase exponentially with the increase in pattern size. For instance, 32 bit IP address needs a 4 GB of RAM and a 72 bit pattern needs 4294967296 TB of RAM, which is practically infeasible in terms of area, cost, and power consumption. It might be a good alternative for CAM in ATM communication systems but cannot be used in applications, which need area, cost, and power as efficiency metrics.

Authors in [6] have proposed a method based on conventional hashing technique to make RAM a content addressable memory. But this method suffers from certain unavoidable disadvantages: collisions and bucket overflow that requires additional area. A perfect hash function is infeasible for random data. The performance of the method greatly depends on hash function. The hash function that minimizes the probability of collision is considered good. But for a random data, which hash function is better, cannot be known in advance.

The alternative for TCAM presented in US patent [7] also suffers from certain Achilles' heels. It is a hybrid approach in

Manuscript received September 23, 2012; revised November 5, 2012. This work was partly supported by the GRRC program of Gyeonggi province. [GRRC Hanyang2010-A01, Developing Sensor Network SoC Equipped with Low Power and Reliability Characteristics].

Zahid Ullah is with the Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, on study leave from FAST-National University of Computer and Emerging Sciences, Peshawar Campus, Pakistan (e-mail: zullah2@student.cityu.edu.hk).

Sanghyeon Baeg is with the Department of Electronic, Electrical, Control, and Instrumentation Engineering, Hanyang University, ERICA Campus, Kyung-Gi-Do, Korea (e-mail: bau@hanyang.ac.kr).

the sense that it mingles CAM and RAM to get overall CAM functionality, thus inheriting the instinctive disadvantages of CAM. In typical TCAM applications, data is totally random, thus grouping would be a very tedious job by finding distinguishing bits. The patent provides a compression scheme but that is well-suited for data having a contiguous ones (1's) or zeros (0's), for example, IP addresses.

CAM of US patent [8] also suffers from the size disadvantage as discussed for US patent [4]. Besides this, it works on data arranged in ascending order and in a real application, data may be randomly distributed. All the TCAM words are to be arranged in ascending order for the method to be applicable. During the ascending arrangement, the original address order is disturbed. If original addresses are taken into consideration, then the memory and power requirements further increase. Original addresses must be stored somewhere else and there is also a need of appropriate partitioning for having an optimized size of TCAM, which are not described by the patent.

Hence, there is a need for SRAM-based TCAM that has the support for storing original addresses and an appropriate partitioning for achieving optimized TCAM, which are included by our VP SRAM-based TCAM.

III. Architecture of VP SRAM-based TCAM

To the best of our knowledge, it is among the first research work on SRAM-based TCAMs. We have introduced a concept of vertical partitioning of conventional TCAM for achieving practical alternative in the form of VP SRAM-based TCAM. Vertical partitioning logically divides conventional TCAM table column-wise into 'n' number of TCAM sub-tables, which are then processed to be stored in their corresponding SRAM blocks. This processing is explained in section IV (A).

Vertical partitioning implies that a TCAM word of width 'W' bits are divided into 'n' sub-words, each of which is of width 'w' bits. A conceptual view of vertical partitioning is illustrated in Fig.1 where a row corresponds to a TCAM word.

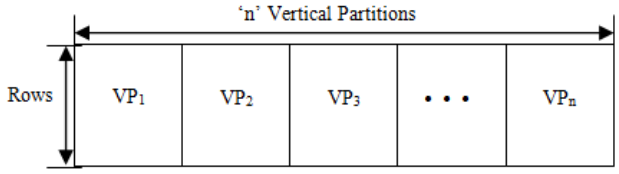

Fig. 1. Conceptual view of vertical partitioning.

Memory architecture of VP SRAM-based TCAM is depicted in Fig. 2. Its main components include 'n' Bit Position Tables (BPTs), 'n' Address Position Tables (APTs), 'n' APT Address Generators (APTAGs), Priority Encoder (PE), and ANDing operation. BPTs and APTs are constructed from SRAM. Each vertical partition has its corresponding BPT, APTAG, and APT.

The maximum possible combinations of 'w' bits are 2^w where each combination represents a sub-word. The aim is to map 2^w sub-words to 2^w bits such that each sub-word would be represented by a single bit in memory.

In BPT, 2^w bits of memory are grouped into 2^{w-b} rows; with each row has 2^b bits, which can be two or more. Each row is supplemented with a value called Last Index (LI) of length w+1 bit.

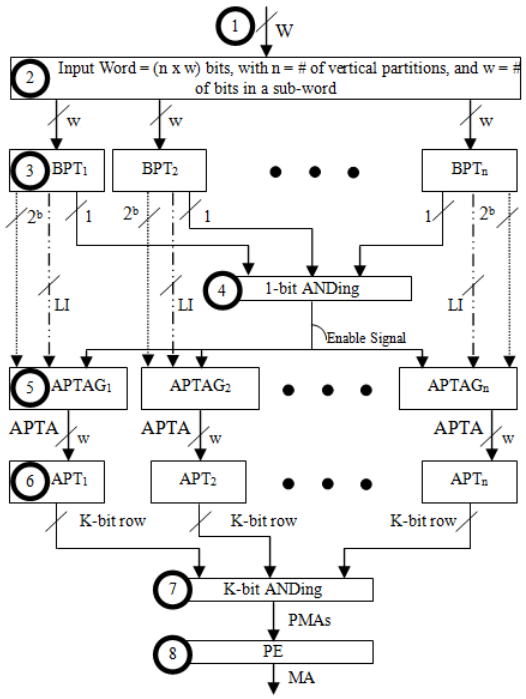

Fig. 2. Memory architecture of VP SRAM-based TCAM.

The 'w-b' high order bits of input sub-word are used to select a particular row in BPT, thus acting as an address. From now onwards this address is termed as BPT Address (BPTA). The 'b' low order bits, from now onwards termed as Bit Position Indicator (BPI), of the input sub-word are used to indicate a particular bit position in the row selected. If the bit position indicated by BPI is high, then it means that the input sub-word is present, otherwise not. Conceptual view of BPT is shown in Fig. 3.

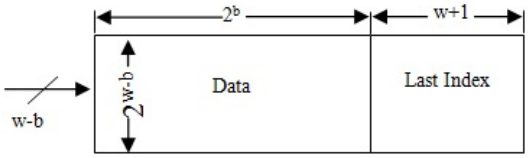

Fig. 3. Conceptual view of BPT.

For instance, if w = 4 bits, then BPT has 2^4 bit positions (0^{th}, 1^{st}, 2^{nd},…, 15^{th}). These sixteen bits, for instance, are arranged in four rows (2^2), with each row having four bits (2^2). LI of a row is set to the total number of bits set in all previous rows reduced by 1. Composition of BPT for the mentioned example is illustrated in Table I where all the bit positions (0^{th} to 15^{th}) are set to 1.

TABLE I: BPT Having All Bit Positions in High Statae

Address	Last index	Bit Positions			
00	-1	0^{th}	1^{st}	2^{nd}	3^{rd}
01	3	4^{th}	5^{th}	6^{th}	7^{th}
10	7	8^{th}	9^{th}	10^{th}	11^{th}
11	11	12^{th}	13^{th}	14^{th}	15^{th}

APTAG generates an address referred to APT Address (APTA), which is used to index a row in APT. APTAG contains 1's counter and adder. The 1's counter counts the number of 1's in the selected row of BPT up to the indicated bit position inclusive and then forwards this information to adder. The adder then adds the output of the 1's counter and LI of the selected row. For example, if input sub-word is 1011, then using Table 1, APTA generated to be 7 + 4 = 11.

The proposed TCAM borrows the concept of BPT and APTAG from [8].

Size of APT is 2^w*K where 2^w represents number of rows and 'K' is the number of bits in each row where each bit represents an address position. This address position corresponds to its original address. It is noteworthy to mention here that 'K' corresponds to rows in Fig. 1. A conceptual view of APT is illustrated in Fig. 4.

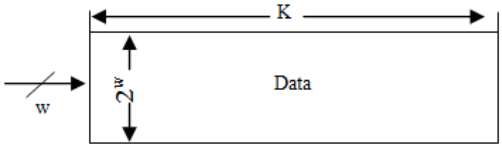

Fig. 4. Conceptual view of APT

An example of APT is shown in Table II where '1' means that its corresponding sub-word in BPT is present at this high (1) address position and '0' means absent.

It should be noted that all BPTs are similar and all APTs are similar.

K-bit rows are read out by using their corresponding APTAs from their corresponding APTs, which are then bit-wise ANDed.

Multiple matches may also occur, which are resolved by PE. PE selects the highest priority matching address as a desired matching address (MA); word in the lowest physical memory location has the highest priority.

TABLE II: AN EXAMPLE OF APT

Address	Address Positions				
	0^{th}	1^{st}	2^{nd}	. . .	$(K-1)^{th}$
0	1	0	0	. . .	0
1	0	1	1	. . .	1
. . .	. . .	. . .	. . .	. . .	. . .
2^{w-1}	1	1	0	. . .	0

IV. PHASES OF VP SRAM-BASED TCAM

A. Data-and-Address Organizing and Storing (DAOS) Phase

In this phase, conventional TCAM table is vertically (column-wise) partitioned, as shown in Fig. 1, into TCAM sub-tables, which are then expanded into binary counter parts and processed in such a way that each sub-word in all partitions is mapped to its corresponding bit in its corresponding BPT and original address (s) of the sub-word are mapped to its/their corresponding bit (s) in the corresponding APT, respectively. Only those bit positions and address positions in BPTs and APTs, respectively, are high, which are mapped while remaining bit positions and address positions are set to low in BPTs and APTs respectively.

After mapping, LIs of their corresponding BPTs are set to their respective values. A conceptual view of mapping of original addresses to their corresponding bits in APT and mapping of sub-words to their corresponding bits in BPT is pictured in Fig. 5

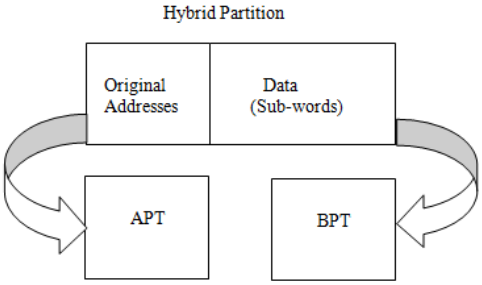

Fig. 5. Mapping of original addresses to their corresponding bits in APT and mapping of sub-words to their corresponding bits in BPT

B. Search Phase

In this phase, an input word is applied and, if exits, its MA is sent to output. The proposed TCAM accomplishes search operation as per steps represented by circles depicted in Fig. 2.

In step 1, an input word is applied to the proposed TCAM. In step 2, the applied word is partitioned into 'n' number of sub-words. These sub-words are then applied to the corresponding SRAM blocks of their corresponding BPTs in parallel. In step 3, bit position in BPT indicated by BPI in the row selected by BPTA is read out. If the read out bit is high (1), it shows that input sub-word is considered present (otherwise, not) but at which address, still unknown. Step 3 occurs in parallel for all BPTs.

In step 4, the read out bits from all BPTs are ANDed. The result of this 1-bit ANDing operation specifies whether the searching effort is to be continued or to be stopped. If the 1-bit ANDing results in a low signal, it means that mismatch has occurred and shows end of search phase, otherwise the proposed TCAM sustains search operation and enters step 5 where APTAG computes APTA by summing the number of one's (1) in the selected row up to the bit position inclusive indicated by BPI and LI of the selected row of BPT. Step 5 is also carried out in parallel for the computation of all the APTAs.

In step 6, APTAs read out rows from their corresponding APTs simultaneously. Step 7 is the K-bit ANDing. The corresponding i^{th} bits of all rows read out in step 6 are ANDed where $0 \leq i \leq K-1$. The address positions that remain at high level (1) after ANDing are considered Potential Matching Addresses (PMAs). In the last step, step 8, a PE selects Matching Address (MA) among the PMAs, if exists, otherwise a mismatch is signaled.

A mismatch of an input word can be signaled at two places; first, when the enable signal is low in step 4 and second, when all the address positions are low (0) after ANDing in step 7.

When 1-bit ANDing of Fig. 2 results in a high output, this condition permits the searching effort to be continued. At this

point, it cannot be decided that the input word is present. It provides only a hope for the match case of the input word. Whenever this case occurs, the match/mismatch is decided by the ANDing operation in step 7 on the accessed rows of the APTs.

V. Analysis of VP SRAM-Based TCAM

Number of vertical partitions has a great impact on the proposed TCAM size. List of symbols for this section is given in Table III.

TABLE III: List of Symbols with Description

Symbols	Description
N_{APT}	Number of transistors in APT
N_{BPT}	Number of transistors in BPT
CT	Conventional TCAM
PT	Proposed TCAM
IF	Increase Factor: the factor by which PT size is increased than its conventional counterpart.
DF	Decrease Factor: the factor by which PT size is decreased than its conventional counterpart.
N_{PT}	Number of transistor in PT
N_{CT}	Number of transistors in CT
TN_{APT}	Total number of transistors in all APTs
TN_{BPT}	Total number of transistors in all BPTs
R_{CT}	Number of rows in CT
Cl_{CT}	Number of columns in CT

Equation (1) shows how to calculate N_{CT}. Equation (2) calculates N_{PT} where N_{APT} and N_{BPT} are explained in (3) and (4). In (1), a typical 16 transistors TCAM cell [3], and in (3) and (4), a typical 6 transistor SRAM cell are taken for analysis.

$$N_{CT} = (R_{CT} \times Cl_{CT}) \times 16 \quad (1)$$

$$N_{PT} = n \times (N_{APT} + N_{BPT}) \quad (2)$$

$$N_{APT} = (2^{w} \times K) \times 6 \quad (3)$$

$$N_{BPT} = ((2^{w-b} \times 2^{b}) + (2^{w-b} \times (w+1))) \times 6 \quad (4)$$

$$IF = \frac{N_{PT}}{N_{CT}} \quad (5)$$

$$DF = \frac{N_{CT}}{N_{PT}} \quad (6)$$

It should be noted here that when NPT > NCT, then (5) is used, otherwise (6) and it should further be noted that routing overhead has not been counted in the size analysis.

Table IV shows a comparison for different number of partitions with respect to size in terms of number of transistors for different conventional TCAM dimensions taken for analysis.

It is evident from Table IV that size of the proposed TCAM has dependency on 'n' and also on the dimensions of the conventional TCAM. In Table IV, DF written with the computed value specifies the reduction of the proposed TCAM by that factor than its traditional counterpart.

Vertical partitioning has dependency on number of 2^b bits in the data portion of BPT (See Fig. 3), which can be two or more bits [6] as explained in section III. Keeping this constraint in mind, maximum possible value of 'n' has been worked out for different types of conventional TCAM dimensions listed in Table IV. It has also been shown that maximum possible value of 'n' offers minimum possible size of VP SRAM-based TCAM. For different dimensions taken in Table IV, it has been analyzed that for maximum possible value of 'n', size of the proposed TCAM is reduced approximately by a factor of 1.3 than its conventional counterpart.

TABLE IV: Size comparison between conventional TCAM and VP SRAM-based TCAM

n	N_{CT}	N_{PT}		DF/IF (Approx.)
		TN_{APT}	TN_{BPT}	
Dimensions = 2^{20} * 72				
36	1207959552	905971824		1.34 (DF)
		905969664	2160	
24	-do-	1207961856		1.00
		1207959552	2304	
18	-do-	1811943216		1.50
		1811939328	3888	
12	-do-	4831850880		4.00
		4831838208	12672	
9	-do-	14495559552		12.00
		14495514624	44928	
Dimensions = 2^{20} * 36				
18	603979776	452985912		1.33 (DF)
		452984832	1080	
12	-do-	603980928		1.00
		603979776	1152	
9	-do-	905971608		1.50
		905969664	1944	
6	-do-	2415925440		4.00
		2415919104	6336	
Dimensions = 512 * 72				
36	589824	444528		1.32 (DF)
		442368	2160	
24	-do-	592128		1.00
		589824	2304	
18	-do-	888624		1.50
		884736	3888	
12	-do-	2371968		4.02
		2359296	12672	
9	-do-	7122816		12.07
		7077888	44928	
Dimensions = 512 * 36				
18	294912	222284		1.32 (DF)
		221184	1080	
12	-do-	296064		1.01
		294912	1152	
9	-do-	444312		1.50
		442368	1944	
6	-do-	1185984		4.02
		1179648	6336	
Dimensions = 512 * 32				
16	262144	197568		1.32 (DF)
		196608	960	
8	-do-	394944		1.51
		393216	1728	
4	-do-	3165696		12.07
		3145728	19968	

It has also been analyzed that generally size of VP SRAM-based TCAM increases exponentially when bits 'w' in sub-word increases (or alternatively when 'n' decreases) as shown in Fig. 6.

Although Fig. 6 is drawn for the size analysis of the proposed TCAM when a conventional TCAM of dimensions 2^{20} * 72 is taken but generally the trend in this graph is followed by all other conventional TCAM dimensions to be mapped to their corresponding proposed TCAM.

In Table IV, memory portion (BPTs and APTs) of the proposed TCAM has been taken for size comparison. Although, size of the proposed TCAM can slightly be increased when 1-bit ANDing for 'n' number of bits and K-bit ANDing for 'n' number of rows each of K bits (See Fig. 2) are considered but still trend shown in Table IV and Fig. 6 is to be followed.

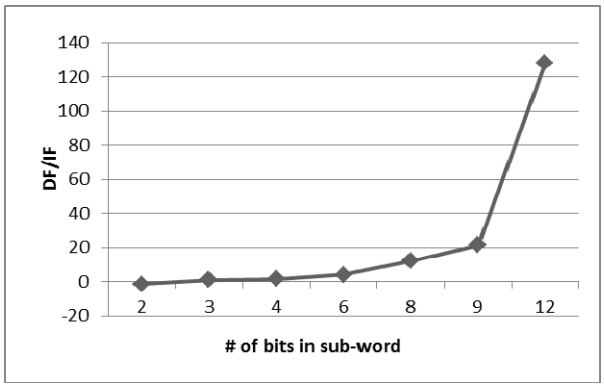

Fig. 6. This graph shows an exponential increase in the memory size of VP SRAM-based TCAM when # of bits in sub-word 'w' increases.

VI. Conclusions

In this paper, we presented VP SRAM-based TCAM that emulates TCAM functionality with SRAM. Vertical partitioning and APT are novel concepts introduced in this research work. Analysis for VP SRAM-based TCAM shows that size and latency are dependent on number of vertical partitions and also on the dimensions of the conventional TCAM. In general, size of VP SRAM-based TCAM increases as the width of sub-word increases or alternatively when number of vertical partitions shrinks.

We analyzed that maximum possible number of vertical partitions provides optimized VP SRAM-based TCAM in terms of important performance parameters concurrently including size and search latency and thus is a practical solution to traditional TCAMs and its variations. It has also been shown that for maximum number of vertical partitions, for the different dimension that we have taken for analysis, size of VP SRAM-based TCAM is smaller approximately by of factor of 1.3 than its corresponding conventional TCAM. Furthermore, VP SRAM-based TCAM ensures large capacities TCAMs whereas this capability is lacked by conventional ones. For analysis, we took typical 16T TCAM cell and 6T SRAM cell.

We are presently working on developing hybrid partitioned SRAM-based TCAM. We are also trying to reveal more methods that organize SRAM in such way that behave like TCAM.

Acknowledgment

The author, Zahid Ullah, is very indebted to the Higher Education Commission (HEC), Government of Pakistan, for the award of MS-Engineering scholarship under the HEC project titled as MS Level Training in Korean Universities/Industry.

References

[1] S. Baeg, "Low Power Ternary Content-Addressable Memory design using a segmented match line," *IEEE Trans. on CAS-I*, vol. 55, no. 6, pp.1485-1494, Jul. 2008.

[2] S. Baeg, S. Wen, and R. Wong, "Minimizing soft errors in TCAM devices: A probabilistic approach to determining scrubbing intervals," *IEEE Trans. on CAS-I*, pp.1-1, June 2009.

[3] K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory (CAM) circuits and architectures: A tutorial and survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp.712-727, March 2006.

[4] S. V. Kartalopoulos, "RAM-based associative Content-Addressable Memory device, method of operation thereof and ATM communication switching system employing the same," US Patent 6097724, Aug. 1, 2000.

[5] P. Mahoney, Y. Savaria, G. Bois, and P. Plante, "Performance characterization for the implementation of Content Addressable Memories based on parallel hashing memories," P. Stenstrom (ED.): *Transactions on HiPEAC II, LNCS 5470*, pp. 307-325, 2009.

[6] S. Cho, J. R. Martin, R. Xu, M. H. Hammoud, and R. Melhem, "CA-RAM: A high-performance memory substrate for search-intensive applications," in *Proceedings of Performance Analysis of Systems & Software, IEEE International Symposium*, pp. 230-241, April, 2007.

[7] M. Somasundaram, "Memory and power efficient mechanism for fast table, lookup," US Patent 7,296,113,B2, Nov. 13, 2007.

[8] M. Somasundaram, "Circuit to generate a sequential index for an input number in a pre-defined list of numbers," US Patent 7155563 B1, Dec. 26, 2006.

Zahid Ullah received the B.Sc. degree (with honors) in Computer System Engineering from the University of Engineering & Technology, Peshawar, Pakistan in 2006 and the M.S. degree (by research) in Electronic, Electrical, Control, and Instrumentation Engineering from the Hanyang University, South Korea in 2010, respectively. He worked as a lecturer in the FAST-National University of Computer and Emerging Sciences, Peshawar Campus, Pakistan for a year. He is currently working toward the Ph.D. degree at Electronic Engineering in City University of Hong Kong. His current research includes SRAM-based architecture for Ternary Content Addressable Memory.

Sanghyeon Baeg received the B.S. degree from Hanyang University, Seoul, Korea in 1986, in Electronic Engineering, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Texas at Austin, Texas, U.S.A in 1988 and 1991, respectively.

From 1994 to 1997 he was a Staff Researcher at Samsung Electronics Co., Kihung, Korea. In 1995, he was dispatched to Samsung Semiconductor Inc. San Jose CA, and worked as a member of the Technical Staff. In 1997, he joined Cisco Systems, Inc. San Jose, CA and worked as a Hardware Engineer, Technical Leader and Hardware Manager. Since 2004, he has been working as Professor at Hanyang University, Ansan, Korea in the School of Electrical Engineering and Computer Science. His work has focused on reliable computing, soft error, low Power CAM, and VLSI DFT implementation and methodologies. He holds many U.S. patents in these fields.

Dr. Baeg received an Inventor Recognition Award from Semiconductor Research Cooperation in 1993. He was an IEEE 1149.6 working group member in 2003. He serves as the president of Institute of Semiconductor Test of Korea from 2012.