Low-Area TCAM Using A Don't Care Reduction Scheme

Ki-Chan Woo and Byung-Do Yang

***Abstract*— This paper proposes a low-area ternary content-addressable memory using a don't care reduction (DCR) scheme. In Internet Protocol (IP) address, the prefix bits store "0" or "1," and the remaining bits store "X" (don't care). The conventional ternary content-addressable memory (TCAM) needs a 2N-bit memory for an N-bit IP address, because its TCAM cell uses 2-bit memory to store a data bit ("0" or "1") and a don't care ("X") bit. However, the proposed DCR TCAM (DCR-TCAM) needs a $(N + \log_2 N)$-bit memory for an N-bit IP address. It also stores N-bit data, but it encodes N-bit "X"s into a $\log_2 N$-bit code, storing the first "X" position. The proposed DCR-TCAM performs the function of "X" in the TCAM from the $\log_2 N$-bit code by using additional decoders and bypass transistors. A 256 × 128-bit DCR-TCAM chip was fabricated using a 1.2-V, 65-nm CMOS process. Its area is 0.22 mm^2, which is only 72% of the conventional TCAM. It expands an effective memory size by applying the data-relocation TCAM scheme. Its energy/bit/search is 0.41 fJ at a clock frequency of 330 MHz.**

***Index Terms*— Don't care, Internet Protocol (IP) address, low area, memory reduction, ternary content-addressable memory (TCAM).**

I. Introduction

TERNARY content-addressable memory (TCAM) quickly searches an input search data from the entire memory and notifies the address of the data in a single clock cycle. Because of the fast search speed, the TCAM is widely used in networking applications such as network routers, lookup tables, and associative memories. Currently, the extensive expansion of the Internet is causing the exponential growth of routing tables in routers, resulting in a large power consumption and high cost [1]–[3].

Fig. 1 shows the conventional TCAM architecture consisting of memory cells, a search word register, a word match circuit, and an address encoder. The TCAM cells are implemented using NAND or NOR CAM cells. They store "0," "1," or "X" (don't care). Thus, each TCAM cell needs 2-bit memory. The TCAM compares the search data on search lines (SLs) with the stored data from all of the TCAM cells to find a match line (ML) storing the search data. The TCAM consumes most of the power in the highly capacitive MLs and SLs.

Manuscript received December 17, 2017; revised March 7, 2018; accepted March 27, 2018. Date of publication April 30, 2018; date of current version July 20, 2018. This paper was approved by Associate Editor Vivek De. This work was supported by the Ministry of Education, Science and Technology through Basic Science Research Program, National Research Foundation of Korea under Grant 2015R1D1A3A01017756. *(Corresponding author: Byung-Do Yang.)*

The authors are with Department of Electronics Engineering, Chungbuk National University, Cheongju 28644, South Korea (e-mail: mykidlove100@cbnu.ac.kr; bdyang@cbnu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2018.2822696

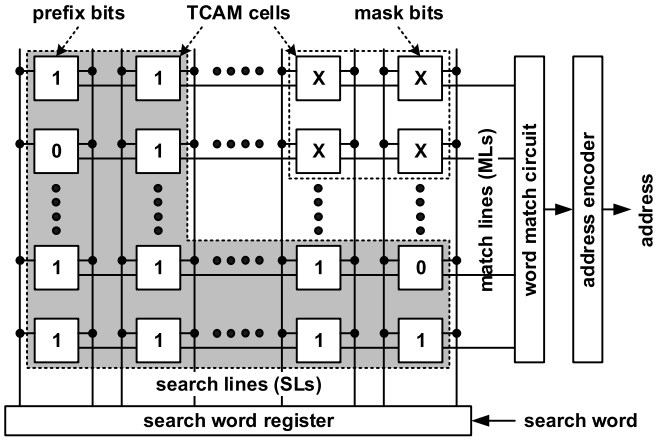

Fig. 1. Conventional TCAM architecture.

The NAND-type CAM consumes the least power in the MLs, although it is slow [3]. In contrast, the NOR-type CAM is fast but it consumes the largest power [4]. To achieve both low-power consumption and high speed, the NOR-type CAMs were developed to reduce the power consumption [5]–[20] and the NAND-type CAMs were developed to improve the searching speed [21]–[25].

In the Internet Protocol version 6 (IPv6), the prefix bits of the IP address store "0" or "1," and the remaining bits (mask bits) store "X," as shown in Fig. 1. The prefix bits with less than 64 bits occupy almost 99% of the 128-bit IP address [23]. Therefore, most of TCAM cells store "X." The data-relocation TCAM (DR-TCAM) expands an effective memory size with low power consumption [1]. The DR-TCAM increases the number of IP addresses stored in the TCAM by relocating the data in the prefix bits into the "X" cells. The DR-TCAM operates at four bank types according to the stored data-bit width. The type-0 bank stores "X" and is inactivated. Type-1 and type-2 banks store four 32-bit words and two 64-bit words instead of a 128-bit word in the type-3 bank, respectively. Therefore, type-1 and type-2 banks store four and two times larger IP addresses, respectively.

In this paper, a low-area TCAM using a don't care reduction (DCR) scheme is proposed. The proposed DCR-TCAM reduces the memory area by encoding N-bit "X"s into a $\log_2 N$-bit code storing the first "X" position. The TCAM cells are replaced with the binary CAM (BCAM) cells that store "0" or "1." The code performs the same function of "X"s by using additional decoders and bypass transistors.

Section II describes the architecture of the proposed DCR-TCAM. Section III shows the measurement results from the fabricated chip. The conclusion is given in Section IV.

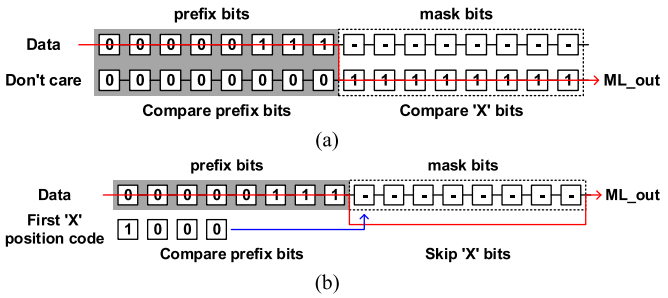

Fig. 2. N-bit IP addresses of (a) conventional TCAM and (b) proposed DCR-TCAM, when $N = 16$.

II. Architecture

A. Concept of Don't Care Reduction TCAM

Fig. 2 shows the N-bit IP addresses of the conventional TCAM and the proposed DCR-TCAM, when $N = 16$. The conventional TCAM needs $2N$-bit memory for an N-bit IP address, because its TCAM cell uses 2-bit memory to store a data bit ("0" or "1") and a don't care ("X") bit. However, the DCR-TCAM needs $(N + \log_2 N)$-bit memory for an N-bit IP address. It also stores N-bit data, but it encodes N-bit"X"s into a $\log_2 N$-bit code storing the first "X" position (first "X" position code). The conventional TCAM compares both the prefix bits and mask bits. On the contrary, the DCR-TCAM skips the mask bits according to the first "X" position code, which performs the same function as that of the N-bit "X"s.

Fig. 3 shows the N-bit AND-ML of the conventional TCAM and the N-bit bypass AND-ML of the DCR-TCAM, when $N = 16$. The conventional TCAM needs N TCAM cells consisting of 2-bit memory cells ($M1$ and $M2$) for a data bit and an "X" bit. However, the DCR-TCAM replaces N "X" cells ($M2$) with a $\log_2 N$-bit code and a thermometer decoder. The TCAM cells are replaced with the BCAM cells that store a data bit. Consequently, the memory cells are reduced from $2N$ bits to $(N + \log_2 N)$ bits, so that the DCR-TCAM has a smaller area than the conventional TCAM despite the extra thermometer decoder.

In the N-bit AND-ML of the conventional TCAM shown in Fig. 3(a), the ML is first precharged to V_{DD} and the ML output (ML_out) becomes "0" when Φ = "0." After Φ is triggered to "1," the search operation starts. In the prefix bits, the series nMOS transistors for data match turn on or off according to the match results between the input search data and the stored data in $M1$. In the mask bits, the series nMOS transistors for "X" match turn on due to "X" stored in $M2$. Therefore, if the input search data matches the stored data in $M1$ and $M2$, the ML pulls down though the series nMOS transistors and the ML_out becomes "1." However, the N-bit bypass AND-ML of the DCR-TCAM replaces $M2$ in the TCAM cells with the first "X" position code and the thermometer decoder in Fig. 3(b). The thermometer decoder outputs TD [1:15] bypass the BCAM cells according to the first "X" position code. In the prefix bits, the search operation of the series nMOS transistors for data match is the same as the conventional AND-ML. However, in the mask bits, the parallel nMOS transistors for ML bypass turn on according to the first "X" position code so that the BCAM cells are bypassed.

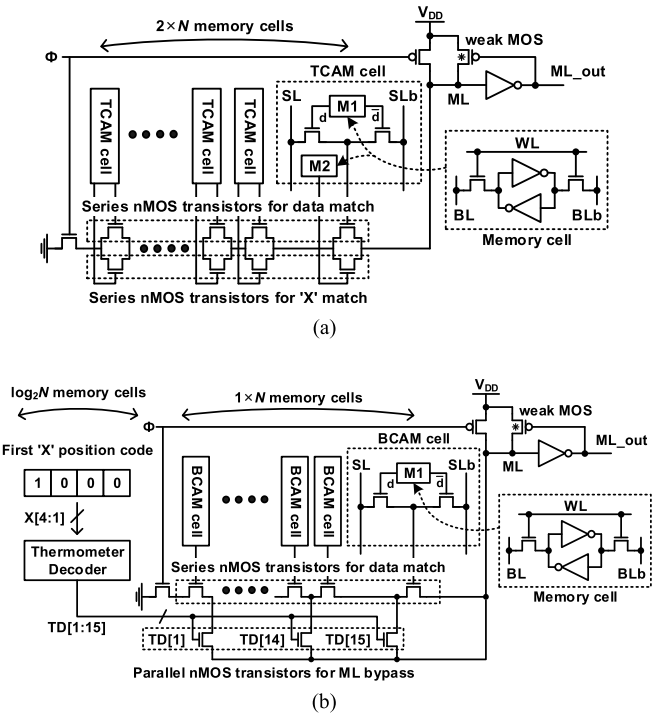

Fig. 3. (a) N-bit AND-ML of the conventional TCAM. (b) N-bit bypass AND-ML of the DCR-TCAM, when $N = 16$.

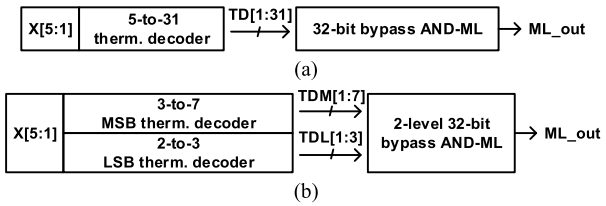

Fig. 4. Block diagrams of a 32-bit DCR-TCAM with (a) conventional thermometer decoder and (b) MSB and LSB thermometer decoders.

B. Architecture of the DCR-TCAM

Fig. 4(a) shows the block diagram of a 32-bit DCR-TCAM consisting of a first "X" position code X[5:1], a 5–31 thermometer decoder, and a 32-bit bypass AND-ML. To bypass the BCAM cells according to the first "X" position code shown in Fig. 3(b), the thermometer decoder is required. The 5–31 thermometer decoder generates 31 output signals TD[1:31] so that the metal lines connected to the 32-bit bypass AND-ML occupy a large area. To reduce the area of the metal lines, multiple metal layers can be used but the metal line connections become more complicated. However, as shown in Fig. 4(b), the number of metal lines can be reduced significantly by using two thermometer decoders of the most significant bits (MSBs) and least significant bits (LSBs). The MSB and LSB thermometer decoders generate 10 output signals consisting of MSB output signals TDM[1:7] and LSB output signals TDL[1:3], which are generated from X[5:3] and X[2:1], respectively, as shown in Fig. 5. In order to bypass the BCAM cells using TDM[1:7] and TDL[1:3], the 32-bit bypass AND-ML is modified to the two-level 32-bit bypass AND-ML. Therefore, the metal lines can be drawn over the BCAM cells so that they need no additional area.

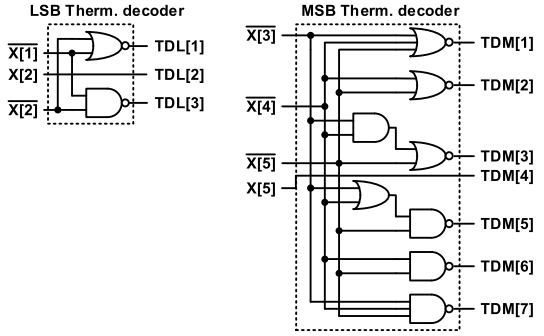

Fig. 5. Thermometer decoders.

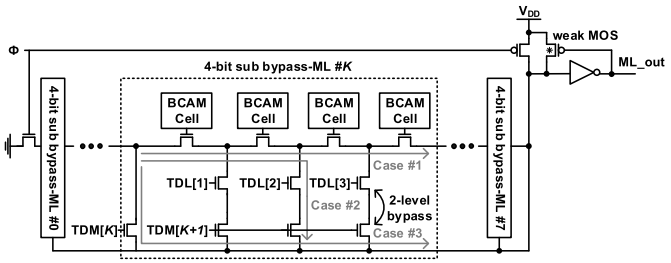

Fig. 6. Two-level 32-bit bypass AND-ML.

Fig. 6 shows the two-level 32-bit bypass AND-ML consisting of eight 4-bit sub-bypass-MLs. The 4-bit sub-bypass-ML has four BCAM cells and the transistors for a two-level bypass. The 4-bit sub bypass-ML #K can operate three cases according to TDM[K] and TDM[$K+1$]. In the case of no bypass, when TDM[K] = "0" and TDM[$K+1$] = "0" (case #1), none of the BCAM cells are bypassed. The two-level bypass is implemented like as a multiplexer by cascading transistors connected to TDL[1:3] and TDM[$K+1$]. If three transistors connected to TDM[$K+1$] are shared with a transistor, undesirable bypass occurs through the transistors connected to TDL[1:3]. In the case of partial bypass, when TDM[K] = "0" and TDM[$K+1$] = "1" (case #2), the BCAM cells are selectively bypassed according to TDL[1:3]. In the case of all bypass MLs, when TDM[K] = "1" (case #3), all BCAM cells are bypassed.

In the first and last 4-bit sub bypass-MLs (#0 and #7), TDM[0] and TDM[8] do not derive from the MSB thermometer decoder. In the first sub ML, TDM[0] = "0" because the first BCAM cell always stores data. In the last sub ML, TDM[8] = "1" because the first BCAM cell stores data when TDM[7] = "0." Table I shows the operation example of the two-level 32-bit bypass AND-ML, when X[5:1] = "'10010." The 4-bit sub bypass-MLs #0–#2 operate in case #1, where no BCAM cells are bypassed. The 4-bit sub bypass-ML #3 operates in case #2, because TDM[3] = "0" and TDM[4] = "1." Two BCAM cells are bypassed when TDL[1:3] = "011." The 4-bit sub bypass-MLs #4–#7 operate in case #3, where all BCAM cells are bypassed.

Table II shows the numbers of transistors of TCAMs. The conventional TCAM (CV-TCAM) has the total number of transistors of 517 in Fig. 3(a). The DCR-TCAM has the total number of transistors of 415 in Fig. 4(b) consisting

TABLE I

OPERATION EXAMPLE OF THE TWO-LEVEL 32-BIT BYPASS AND-ML WHEN X[5:1] = "10 010"

	Case 1			Case 2	Case 3			
X[5:1]	10010							
4-bit sub ML	#0	#1	#2	#3	#4	#5	#6	#7
'X' bits	0000	0000	0000	0011	1111	1111	1111	1111
TDM[1:7]	-	0	0	0	1	1	1	1
TDL[1:3]	011	011	011	011	011	011	011	011

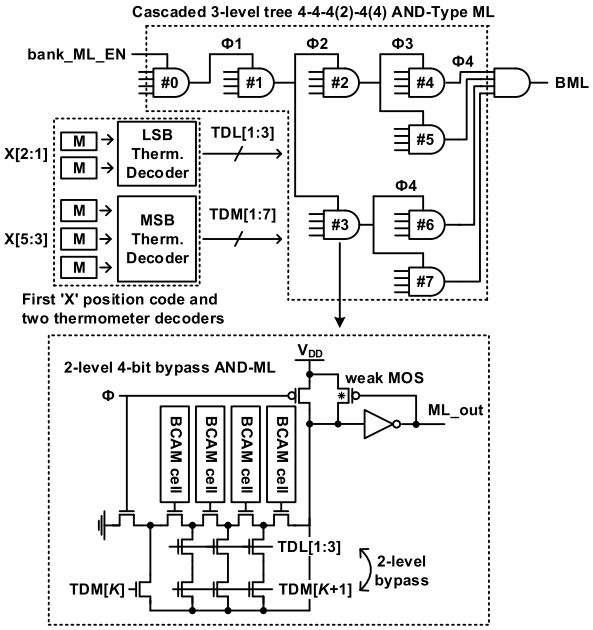

Fig. 7. 32-bit DCR-TCAM using the cascaded three-level tree 4-4-4(2)-4(4) AND-ML.

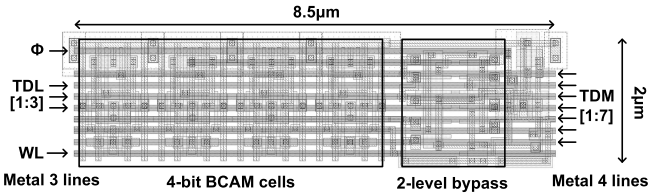

Fig. 8. Layout of the two-level 4-bit bypass AND-ML.

of the circuits in Figs. 5 and 6. Therefore, the DCR-TACM decreases the number of transistors by 20% compared with the CV-TCAM. The 32-bit DCR-TCAM uses the cascaded three-level tree 4-4-4(2)-4(4) AND-type ML [21] in the consideration of the performance, power, area, and stability, as shown in Fig. 7. The 32-bit DCR-TCAM consists of eight two-level 4-bit bypass AND-MLs, the first "X" position code, and two thermometer decoders.

Fig. 8 shows the layout of the two-level 4-bit bypass AND-ML. It uses four metal layers to make the 32-bit DCR-TCAM in Fig. 7. Metal 1 lines are used for local connections. Metal 2 lines are used for SLs and bit lines. Metal 3 lines are used for TDL[1:3] and global connections. Metal 4 lines are used for TDM[1:7] to connect each of eight two-level 4-bit bypass AND-MLs.

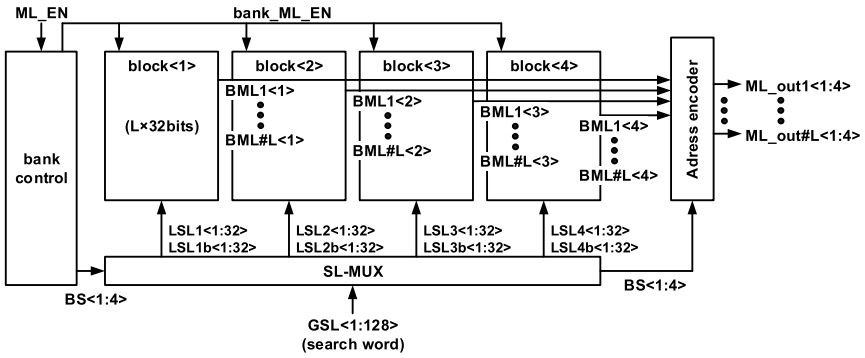

Fig. 9. Bank structure of the DCR-TCAM [1].

TABLE II
NUMBERS OF TRANSISTORS OF CV-TCAM AND DCR-TCAM

		CV-TCAM	DCR-TCAM
Bit length		32-bit	
Number of Transistors	TCAM	448	-
	BCAM	-	256
	X[5:1]	-	30
	Series nMOS transistors	64	32
	2-level bypass	-	56
	Thermometer decoders	-	36
	Passing gate controlled by Φ	5	5
	Total	517	415 (80%)

TABLE III
COMPARISONS OF AREAS OF TCAMS

			CV-TCAM	DR-TCAM	DCR-TCAM
Configuration			M×128-bits		
Bank size			32×128-bits		
Prefix length			1~128		
Area [μm^2]	Control unit & I/O circuits		26,344		
	bank	TCAM cells	34,240		-
		BCAM cells & First 'X' position code	-		22,016
		WL decoder	345		
		SL-MUXs (SL buffers)	382	1,106	
		ML selector	-	614	
		Address encoder	587	649	
		Bank control	-	28	
		Total	35,554	36,919 (104 %)	24,695 (70 %)
Total area (CNT, I/O, banks) (A) [μm^2]		M=256 (8 banks)	310,779	321,695 (103.7%)	223,905 (72%)
		M=4K (128 banks)	4,577,302	4,751,952 (104%)	3,187,316 (69.8%)
Maximum number of stored words (B)*		M=256	256	672 (263%)	
		M=4K	4,096	12,064 (295%)	
Effective area/word (A/B) [μm^2]		M=256	1,214	479 (49.5 %)	333 (27.5%)
		M=4K	1,118	394 (35.3 %)	264 (23.7%)

* Maximum number of stored words is calculated with the statistic prefix length distribution of IP addresses [1].

Fig. 9 shows the bank structure of the DCR-TCAM, which adopts the DR-TCAM [1]. It consists of four $L \times 32$ bit cell blocks (block⟨1:4⟩), a bank control, a SL multiplexer (SL-MUX), and an address encoder. The DCR-TCAM operates at four bank types according to the stored data-bit width. The bank type changes according to four block select signals (BS⟨1:4⟩) in the bank control. The SL-MUX selectively connects four local SLs (LSLs) from LSL1 to LSL4 with the global SLs (GSL). The address encoder selectively connects four block MLs (BML⟨1:4⟩) with four ML outputs (ML_out⟨1:4⟩). Therefore, the DCR-TCAM can significantly store data than the conventional TCAM by applying the data-relocation scheme in the DR-TCAM [1]. Moreover, it reduces the area by using the proposed DCR scheme.

TABLE IV
FEATURES OF THE DCR-TCAM CHIP

Technology	65nm CMOS
Supply voltage	1.2V
Max. clock freq.	330MHz
Organization	256×128bit cells (8 banks)
Chip core area	0.22mm^2 (0.71mm×0.31mm)
Energy/bit/search	0.41fJ

The DCR-TCAM changes the bank type according to the data-bit width. It stores 1–32-bit data, 33–64-bit data, and 65–128-bit data in a block, two blocks, and four blocks, respectively. Therefore, the first BCAM cells from block⟨1⟩ to block⟨3⟩ always store data, so that TDM[0] = "0" to avoid bypassing the blocks. However, when 96-bit data are stored, the first BCAM cell of block⟨4⟩ has "X," but block⟨4⟩ is not bypassed because TDM[0] = "0." Block⟨4⟩ can be bypassed by ignoring the 4th BMLs (BML⟨4⟩) with a memory cell in the address encoder.

Fig. 10 shows the block diagram of the 256×128-bit DCR-TCAM, which adopts the DR-TCAM [1]. It consists of eight banks, a control unit, an input/output (I/O) circuits, and 10-bit DFFs. Each bank is the same bank structure shown in Fig. 9, when $L = 32$. The DCR-TCAM was implemented in a 65-nm CMOS process at $V_{DD} = 1.2$ V.

Fig. 11 shows the simulated waveforms of the DCR-TCAM at the worst delay condition. In order to measure the speed and power consumption including the parasitic parameters, the post-layout simulation is performed. The SL driving time is 0.7 ns, which is the total time of the delay from the clock signal to the GSL signal (0.2 ns), the delay from the GSL signal to the LSL signal (0.2 ns), and the timing margin between the LSL signal and the bank_ML_EN signal (0.3 ns). The ML comparing time is 1.8 ns, which is the delay from the bank_ML_EN to the BML, as shown in Fig. 7. The address encoding time is 0.3 ns, which is the delay from the

TABLE V
PERFORMANCE COMPARISONS OF TCAMS

	TCAS1 2011 [5]	JSSC 2013 [13]	JSSC 2013 [18]	JSSC 2008 [21]	JSSC 2011 [23]	JSSC 2015 [1]		This work	
Technology	180nm CMOS	65nm CMOS	32nm SOI	180nm CMOS	65nm CMOS	130nm CMOS	65nm CMOS *1	65nm CMOS	
V_{DD}	1.8V	1.0V	0.95V	1.8V	1V	1.2V		1.2V	
CAM type	NOR TCAM	NOR TCAM	NOR TCAM	NAND TCAM	NAND TCAM	NAND TCAM		NAND TCAM	
Configuration (Total CAM bits)	128×144b	18Mb (128K×144b)	2K×640b (10K×128b)	256×128b	256×144b	256×128b	256×128b	256×128b	4K×128b
Chip area [mm^2]	0.98	74	1.56	0.97	0.43	0.87	0.32	0.22	3.19
Max. number of stored words *2	128	128K	10K	256	256	672	672	672	12,064
Effective area/word [μm^2] *3	7,656	578	156	3,789	1,680	1296	476	327	264
Effective area/word [μm^2] (Normalized to 65nm) *4	998	578	644	494	1680	324	476	327	264
Energy/bit/search [fJ]	2.82	1.98	0.58	1.42	0.165	1.3	0.41	0.41	0.27
Effective energy/word/search [fJ] *5	406	285	74	182	24	63	20	20	11
Effective energy/word/search [fJ] (Normalized to 65nm) *6	65	411	241	29	34	16	20	20	11
Max. clock freq. [MHz]	210	250	1,000	320	400	200	330	330	330

*1 For comparisons, the DR-TCAM is applied to 65nm CMOS process.
*2 For the calculation of the maximum number of stored words, the statistic prefix length distribution of IP addresses is used [23].
*3 Effective area/word = chip area / the maximum number of stored words [1].
*4 For normalization for area, the factor of (65/Technology)^2 is multiplied [13].
*5 Effective energy/word/search = (Energy/bit/search) × total CAM bits / the maximum number of stored words [1].
*6 For normalization for energy/word/search, the factor of (65/Technology) × (1.2/V_{DD})^2 is multiplied [13].

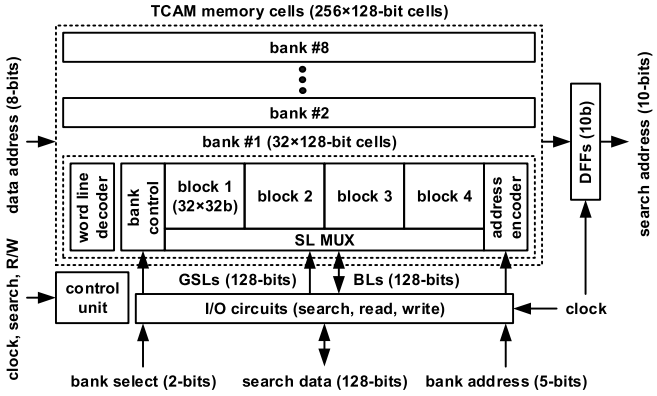

Fig. 10. Block diagram of the 256 × 128-bit DCR-TCAM [1].

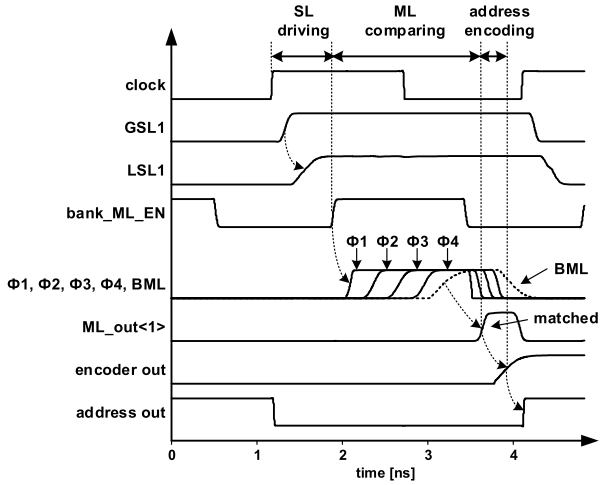

Fig. 11. Simulated waveforms of the DCR-TCAM.

ML_out⟨1⟩ signal to the encoder output signal. The timing margin is 0.2 ns to latch encoder output signal in the 10-bit DFFs. Therefore, the clock cycle time is required at least 3 ns and the maximum operation clock frequency is 330 MHz.

C. Comparisons of the DCR-TCAM

Table III shows the area comparisons of TCAMs. The TCAMs have $M \times 128$-bit cells. Each bank has 32×128-bit cells. The three TCAMs have the same area in the control unit and I/O circuit. The bank of the conventional TCAM (CV-TCAM) consists of TCAM cells, a word line (WL) decoder, SL buffers, and an address encoder. Both the DR-TCAM and DCR-TCAM need more areas for SL-MUXs, an ML selector, and a bank control circuit than the CV-TCAM. The bank area of DR-TCAM is 4% larger than that of the CV-TCAM.

However, the DCR-TCAM replaces the TCAM cells with the BCAM cells and the first "X" position code. The bank area of the DCR-TCAM is 30% smaller than that of the CV-TCAM. The total areas of the DCR-TCAM are 28% and 30% smaller than those of the CV-TCAM at $M = 256$ and $M = 4$ K, respectively. The maximum number of stored words is calculated with the statistic prefix length distribution of IP addresses [1]. As a result, the effective word area (area/word = total area of the TCAM/the maximum number of stored words) of the DCR-TCAM is the smallest of the TCAMs. The effective word areas of the DCR-TCAM are 27.5% and 23.7% of those of the CV-TCAM at $M = 256$ and $M = 4$ K, respectively.

III. EXPERIMENTAL RESULTS

Fig. 12 shows the chip microphotograph of the 256×128 bit DCR-TCAM chip. The DCR-TCAM features list in Table IV. The DCR-TCAM was fabricated in a 65-nm CMOS process with $V_{DD} = 1.2$ V. Its area is 0.22 mm^2. Its energy/bit/search is 0.41 fJ at a maximum clock frequency of 330 MHz. Fig. 13 shows the maximum clock frequency according to V_{DD}.

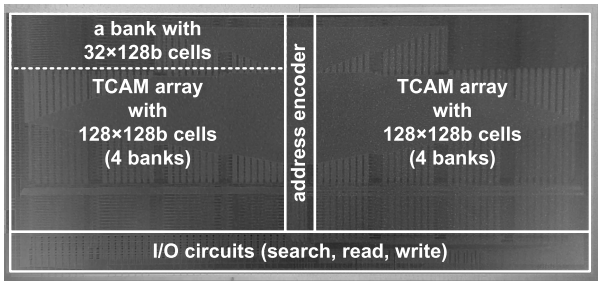

Fig. 12. Microphotograph of the DCR-TCAM chip.

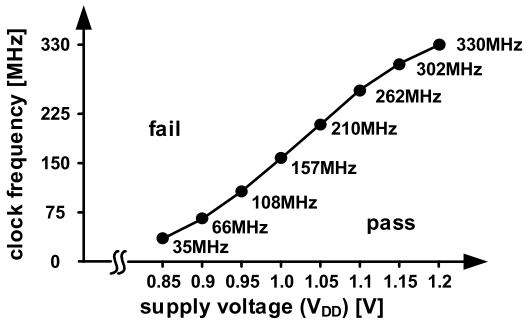

Fig. 13. Maximum clock frequency versus V_{DD}.

Table V shows the performance comparisons of TCAMs. In order to compare performances, the effective area/word and the effective energy/word/search are added. The effective area/word (=chip area over the maximum number of stored words) shows the chip area per a stored word. The effective energy/word/search (=energy/bit/search × total CAM bits/the maximum number of stored words) shows the enery per a stored word and a search. The effective area/word is normalized to 65-nm technology by multiplying the factor of $(65/\text{Technology})^2$ [13]. The effective energy/word/search is normalized to 65-nm technology by multiplying the factor of $(65/\text{Technology}) \times (1.2/V_{DD})^2$ [13]. The DCR-TCAM and DR-TCAM store the same maximum number of stored words. However, the DCR-TCAM has a smaller area than the DR-TCAM by using the DCR scheme. Thus, the DCR-TCAM has the smallest effective area/word.

IV. Conclusion

In this paper, the low-area TCAM using the DCR scheme is proposed. The conventional TCAM needs $2N$-bit memory cells for an N-bit IP address, because its TCAM cell uses 2-bit memory to store the data bit and the "X" bit. However, the DCR-TCAM encodes N-bit "X"s into the $\log_2 N$-bit code that stores the first "X" position, which performs the function of "X"s in the TCAM by using additional decoders and bypass transistors. The DCR-TCAM replaces the TCAM cells with the BCAM cells. Therefore, it needs $(N+\log_2 N)$-bit memory cells for an N-bit IP address. A 256 × 128-bit DCR-TCAM chip was fabricated using a 1.2-V, 65-nm CMOS process. Its area was 0.22 mm^2, which is only 72% of the conventional TCAM. The DCR-TCAM expands the effective memory size by applying the DR-TCAM scheme [1]. Its energy/bit/search is 0.41 fJ at a clock frequency of 330 MHz.

Acknowledgment

The chip fabrication was supported by the IC Design Education Center.

References

[1] B.-D. Yang, "Low-power effective memory-size expanded TCAM using data-relocation scheme," *IEEE J. Solid-State Circuits*, vol. 50, no. 10, pp. 2441–2450, Oct. 2015.

[2] C.-C. Chuang, Y.-J. Yu, A.-C. Pang, and G.-Y. Chen, "Minimization of TCAM usage for SDN scalability in wireless data centers," in *Proc. IEEE Global Commun. Conf.*, Dec. 2016, pp. 1–7.

[3] R. B. Kadir and A. S. Anwar, "Analysis of charge-shared matchline sensing schemes and current race scheme in high-speed ternary content addressable memory (TCAM)," in *Proc. IEEE Int. Conf. Innov. Sci., Eng. Technol.*, Oct. 2016, pp. 1–4.

[4] F. Shafai, K. J. Schultz, G. F. R. Gibson, A. G. Bluschke, and D. E. Somppi, "Fully parallel 30-MHz, 2.5-Mb CAM," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1690–1696, Nov. 1998.

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

[6] H. Miyatake, M. Tanaka, and Y. Mori, "A design for high-speed low-power CMOS fully parallel content-addressable memory macros," *IEEE J. Solid-State Circuits*, vol. 36, no. 6, pp. 956–968, Jun. 2001.

[7] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.

[8] I. Arsovski and A. Sheikholeslami, "A mismatch-dependent power allocation technique for match-line sensing in content-addressable memories," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1958–1966, Nov. 2003.

[9] N. Mohan, W. Fung, D. Wright, and M. Sachdev, "A low-power ternary CAM with positive-feedback match-line sense amplifiers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 3, pp. 566–573, Mar. 2009.

[10] C.-C. Wang, C.-H. Hsu, C.-C. Huang, and J.-H. Wu, "A self-disabled sensing technique for content-addressable memories," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 31–35, Jan. 2010.

[11] N. Mohan and M. Sachdev, "Low-capacitance and charge-shared match lines for low-energy high-performance TCAMs," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 2054–2060, Sep. 2007.

[12] S. Baeg, "Low-power ternary content-addressable memory design using a segmented match line," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1485–1494, Jul. 2008.

[13] C.-S. Lin *et al.*, "A 250-MHz 18-Mb full ternary CAM with low-voltage matchline sensing scheme in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2671–2680, Nov. 2013.

[14] A. T. Do, C. Yin, K. Velayudhan, Z. C. Lee, K. S. Yeo, and T. T.-H. Kim, "0.77 fJ/bit/search content addressable memory using small match line swing and automated background checking scheme for variation tolerance," *IEEE J. Solid-State Circuits*, vol. 49, no. 7, pp. 1487–1498, Jul. 2014.

[15] C.-S. Lin, J.-C. Chang, and B.-D. Liu, "A low-power precomputation-based fully parallel content-addressable memory," *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 654–662, Apr. 2003.

[16] K. Pagiamtzis and A. Sheikholeslami, "A low-power content-addressable memory (CAM) using pipelined hierarchical search scheme," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1512–1519, Sep. 2004.

[17] B.-D. Yang and L.-S. Kim, "A low-power CAM using pulsed NAND-NOR match-line and charge-recycling search-line driver," *IEEE J. Solid-State Circuits*, vol. 40, no. 8, pp. 1736–1744, Aug. 2005.

[18] I. Arsovski, T. Hebig, D. Dobson, and R. Wisort, "A 32 nm 0.58-fJ/bit/search 1-GHz ternary content addressable memory compiler using silicon-aware early-predict late-correct sensing with embedded deep-trench capacitor noise mitigation," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 932–939, Apr. 2013.

[19] Y. J. Chang, "A high-performance and energy-efficient TCAM design for IP-address lookup," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 6, pp. 479–483, Jun. 2009.

[20] S. Hanzawa, T. Sakata, K. Kajigaya, R. Takemura, and T. Kawahara, "A large-scale and low-power CAM architecture featuring a one-hot-spot block code for IP-address lookup in a network router," *IEEE J. Solid-State Circuits*, vol. 40, no. 4, pp. 853–861, Apr. 2005.

[21] B.-D. Yang, Y.-K. Lee, S.-W. Sung, J.-J. Min, J.-M. Oh, and H.-J. Kang, "A low power content addressable memory using low swing search lines," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 12, pp. 2849–2858, Dec. 2011.
[22] H.-Y. Li, C.-C. Chen, J.-S. 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.
[23] C.-C. Wang, J.-S. 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.
[24] C. C. Wang, C. J. Cheng, T. F. Chen, and J. S. Wang, "An adaptively dividable dual-port BiTCAM for virus-detection processors in mobile devices," *IEEE J. Solid-State Circuits*, vol. 44, no. 5, pp. 1571–1581, May 2009.
[25] P.-T. Huang and W. Hwang, "A 65 nm 0.165 fJ/bit/search 256×144 TCAM macro design for IPv6 lookup tables," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 507–519, Feb. 2011.
[26] Y.-D. Kim, H.-S. Ahn, S. Kim, and D.-K. Jeong, "A high-speed range-matching TCAM for storage-efficient packet classification," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 6, pp. 1221–1230, Jun. 2009.
[27] J.-R. Yuan, C. Svensson, and P. Larsson, "New domino logic precharged by clock and data," *Electron. Lett.*, vol. 29, no. 25, pp. 2188–2189, Dec. 1993.

Ki-Chan Woo received the B.S. degree in electronics engineering from Chungbuk University, Cheongju, South Korea, in 2013, where he is currently pursuing the Ph.D. degree in electrical engineering and computer Science.

His current research interests include analog circuits, digital circuits, memory circuits, and power IC designs.

Byung-Do Yang received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1999, 2001, and 2005, respectively.

In 2005, he joined the Memory Division, Samsung Electronics, Kyungki-do, South Korea, as a Senior Engineer, where he was involved in the design of DRAM. In 2006, he joined the Department of Electronics Engineering, Chungbuk National University, Cheongju, South Korea, where he is currently a Full Professor. His current research interests include analog circuits, digital circuits, memory circuits, and power IC designs.