Synthesis and Design of Parameter Extractors for Low-Power Pre-computation-Based Content-Addressable Memory Using Gate-Block Selection Algorithm

臺科大電子

Jui-Yuan Hsieh

Outline

Introduction

- Content-addressable memory (CAM) is a storage device, which provides an efficiently fast data-search function.

- To achieve an effective function of data searching, the data comparison architecture of CAMs is usually implemented in parallel operation structure.

Introduction

- However, due to parallel process characteristic, power consumption is always an important concern when designing CAM circuitry.

- Therefore, many articles have been devoted to the study of CAMs for low-power, in which power reduction has focused on the circuit and architecture domains.

Previous Work & Observation

- Recently, pre-computation technique has received as one of the most effective approaches for low-power designs.

- Pre-computation-Based CAM (PB-CAM) stores extra information along with data used in the data searching operation to eliminate most of the unnecessary comparison operations, thereby saving power.

Previous Work & Observation

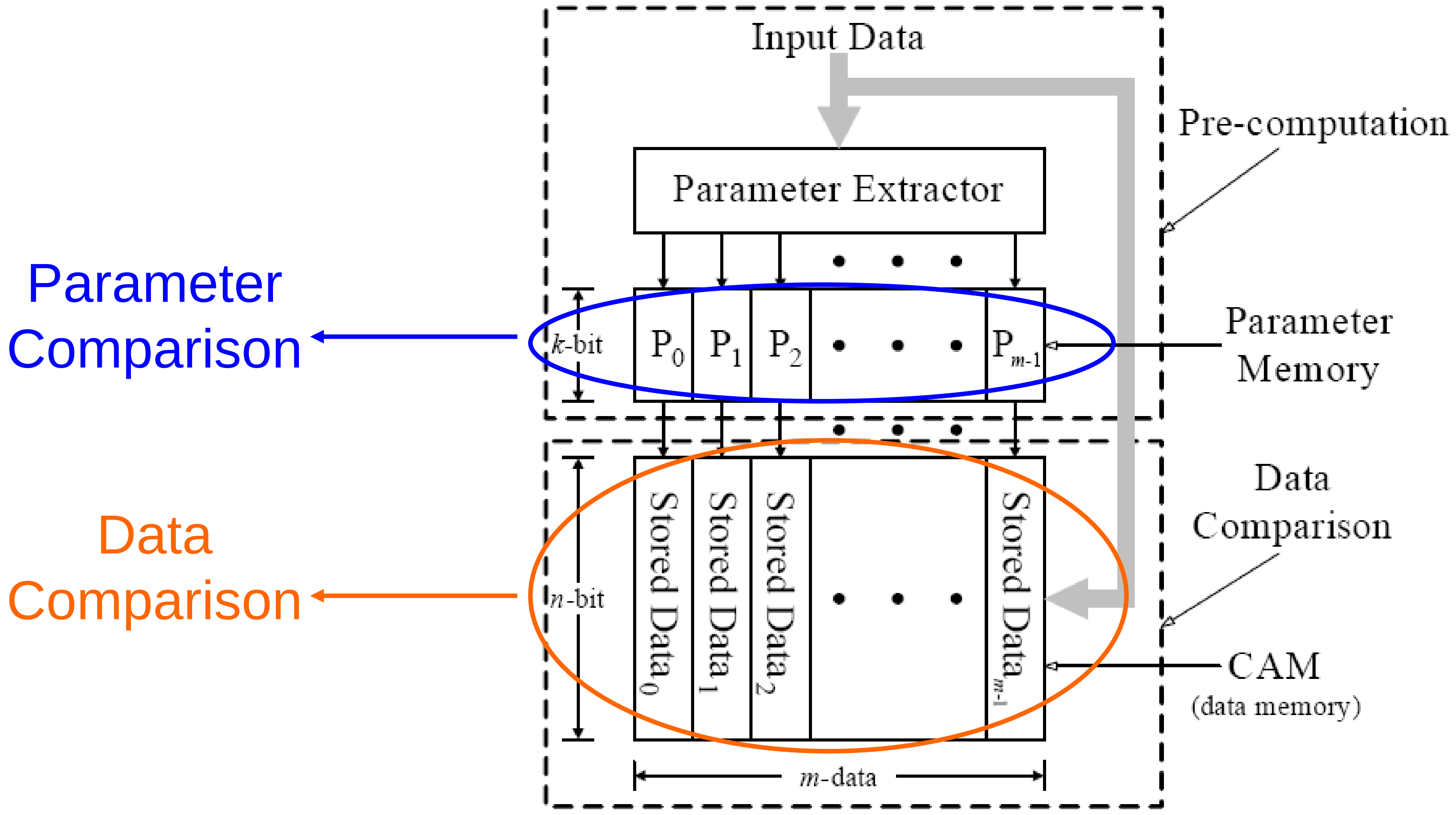


Fig. 1. Memory organization of the PB-CAM.

1’s Count PB-CAM

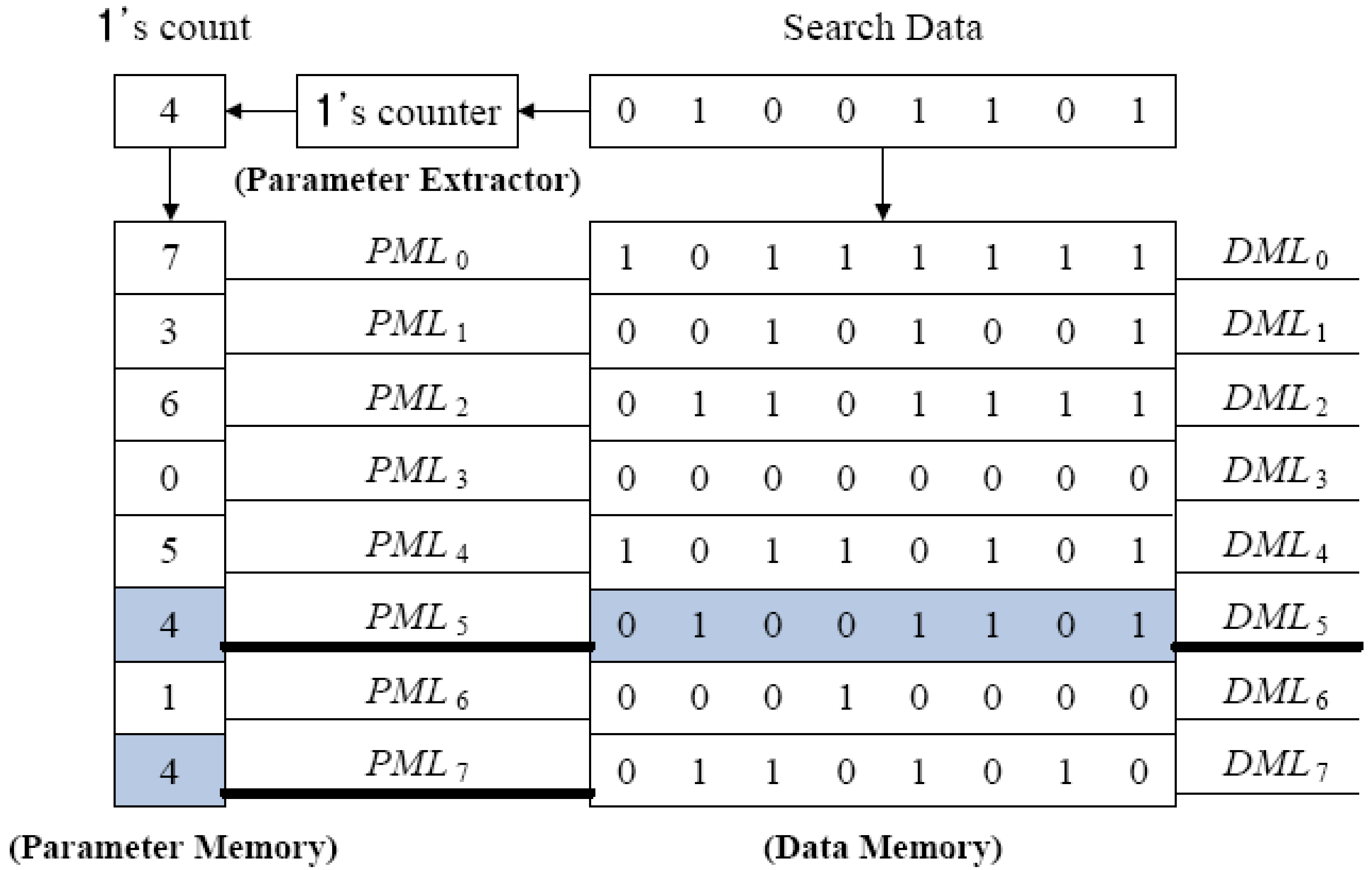


Fig. 2. Conceptual view of the 1’s count PB-CAM.

Mathematical Analysis for the 1's Count PB-CAM

- Assume that the inputs are independent and uniformly distributed, the number of input data related to the same parameter for *n*-bit input data can be determined by

$$N_{s.p.} = \binom{n}{k} = \frac{n!}{k!(n-k)!} \quad ,$$

where *k* is a number of ones for *n*-bit input data (which is from 0 to *n*).

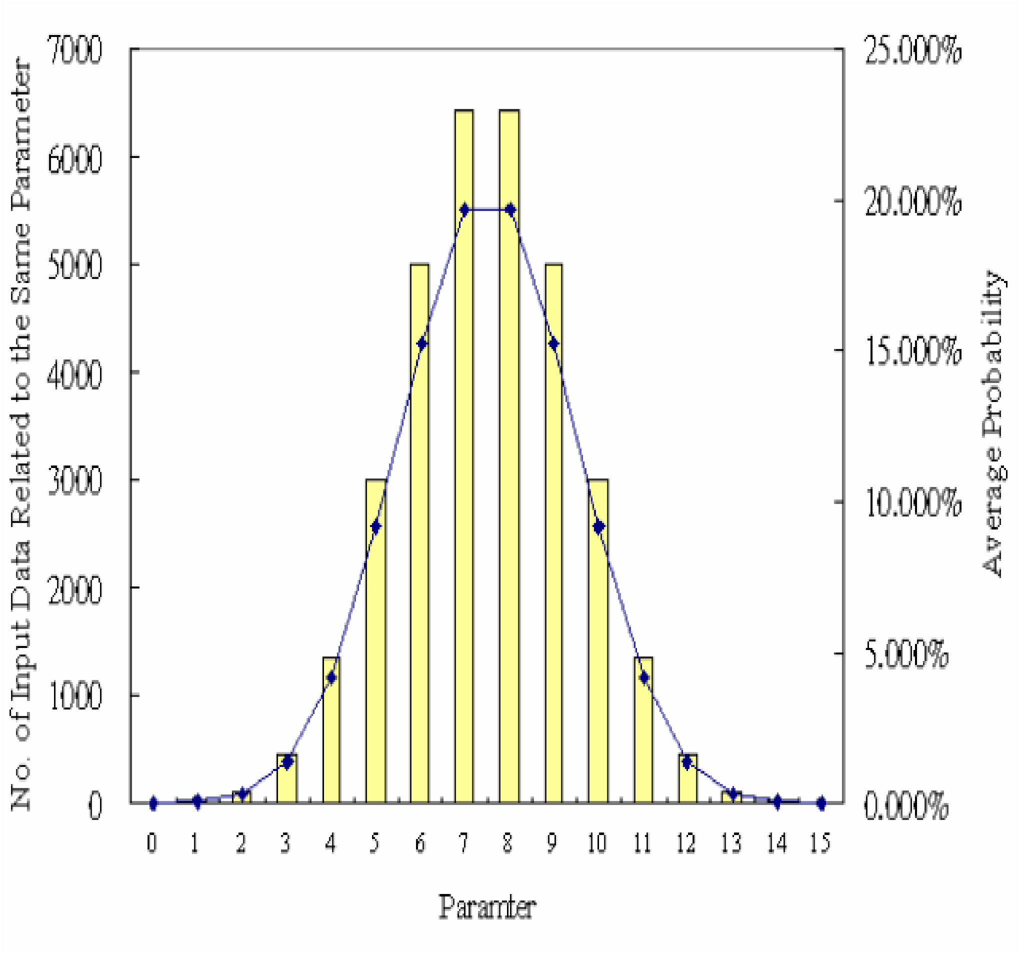
- Therefore, the occurring average probability of every parameter can be derived by

$$P_{avg} = \frac{N_{s.p.}}{2^n} \quad .$$

Mathematical Analysis for the 1's Count PB-CAM

MATHEMATICAL FEATURE OF THE 1'S COUNT APPROACH FOR 15-BIT RANDOM INPUT DATA

Parameter		Number of Data Related to the Same Parameter	Average Probability
0	0000	1	0.003%
1	0001	15	0.046%
2	0010	105	0.320%
3	0011	455	1.389%
4	0100	1365	4.166%
5	0101	3003	9.164%
6	0110	5005	15.274%
7	0111	6435	19.638%
8	1000	6435	19.638%
9	1001	5005	15.274%
10	1010	3003	9.164%
11	1011	1365	4.166%
12	1100	455	1.389%
13	1101	105	0.320%
14	1110	15	0.046%
15	1111	1	0.003%



1's Count Parameter Extractor

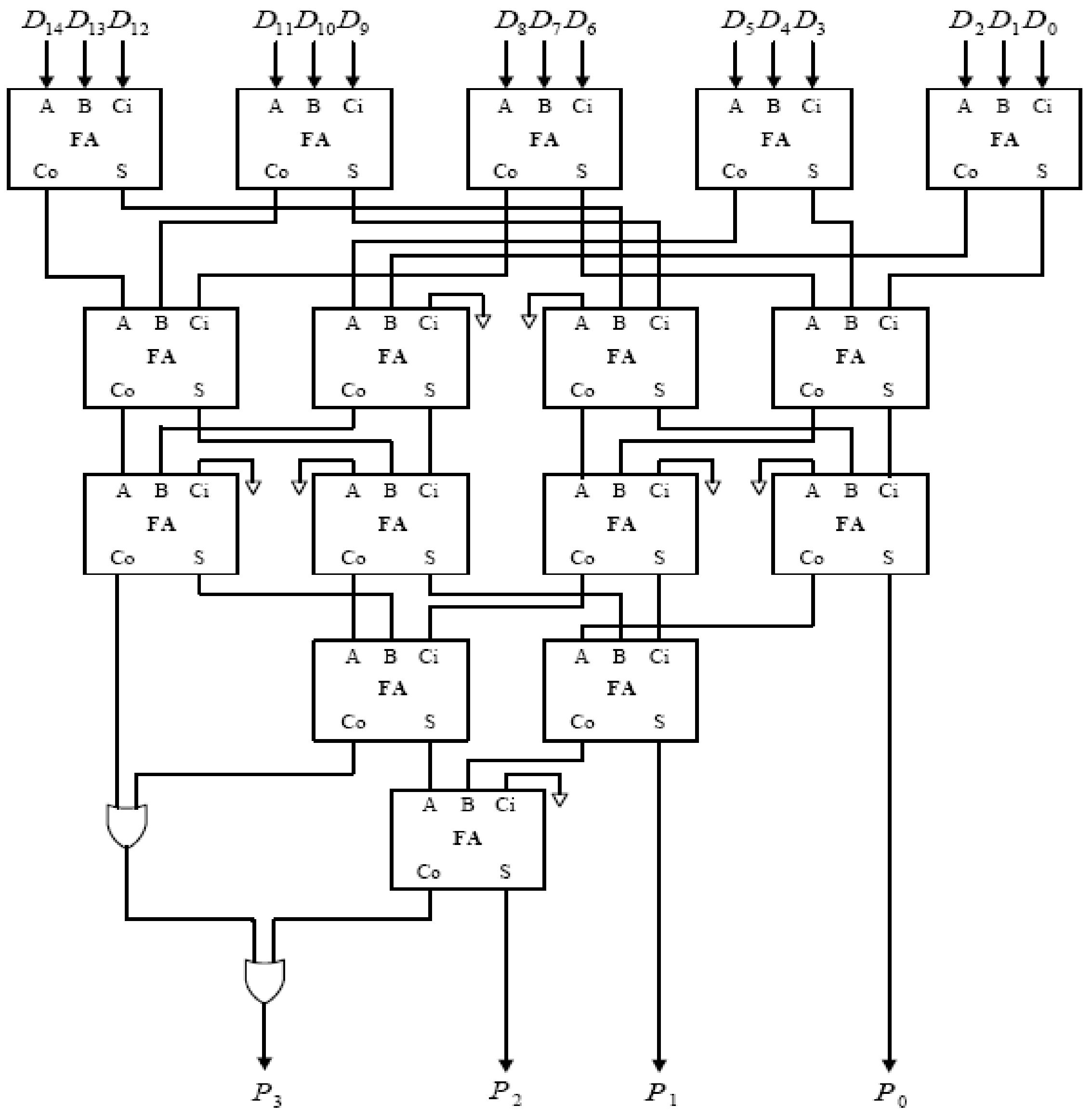


Fig. 5. 15-bit parameter extractor of the 1's count PB-CAM.

Deficiencies of the 1’s Count Approach

1. The normal distribution limits further the reduction of comparison operations in PB-CAMs so that the effect of reducing power consumption is also limited.

2. The 1’s count parameter extractor is implemented with many full adders, which not only wastes area but increases delay.

Block-XOR PB-CAM

- To improve the major deficiencies of the 1’s count approach, the concept behind the block-xor approach is to eliminate the normal distribution characteristic for random input data resulting from the 1’s count approach and to reduce the delay and area of the parameter extractor.

Block-XOR Parameter Extractor

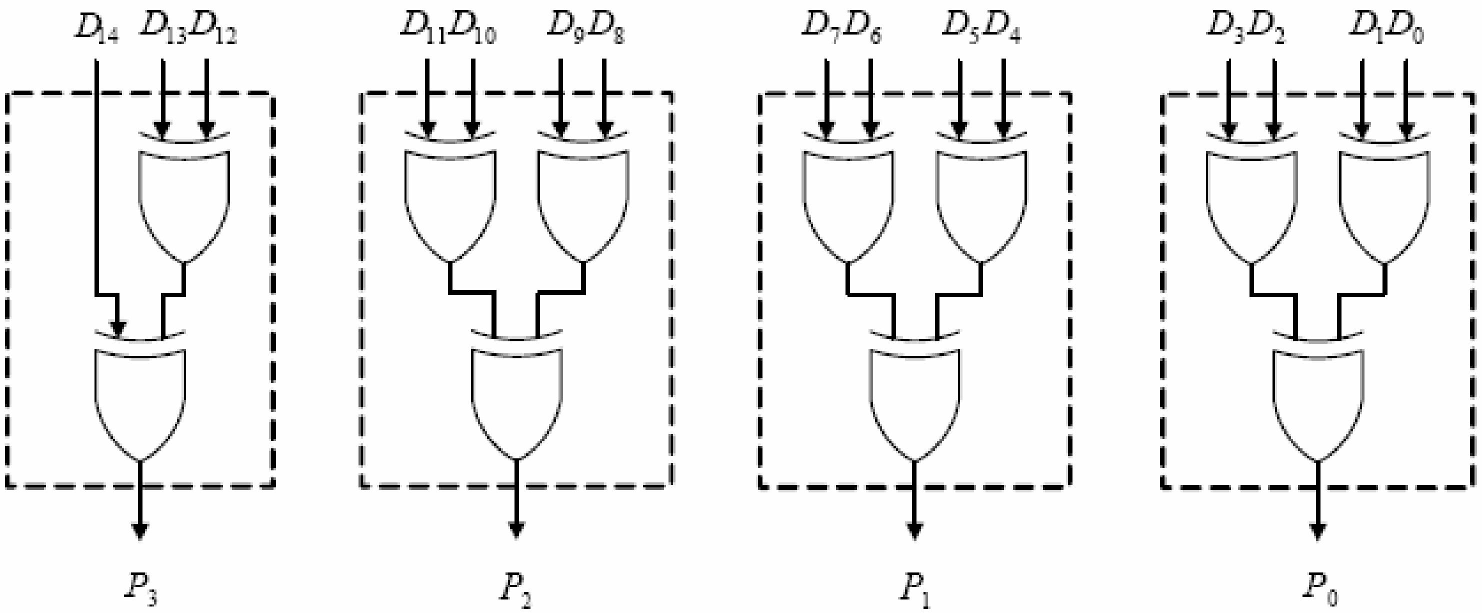
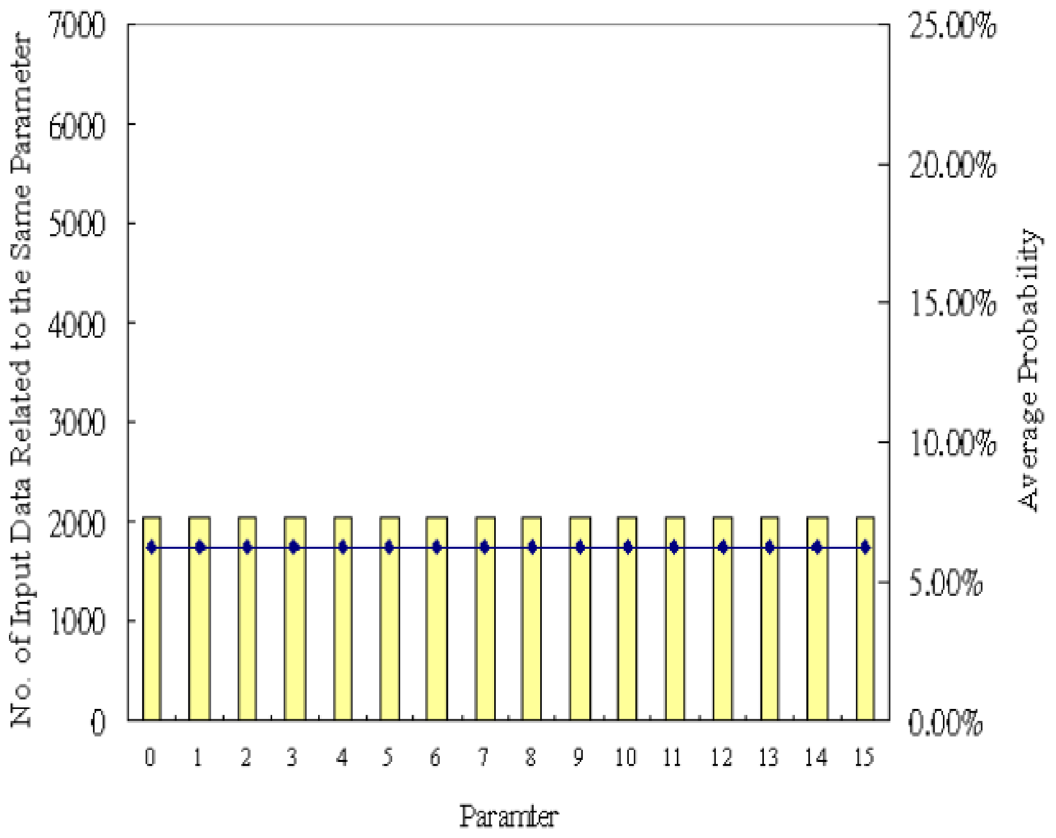


Fig. 6. 15-bit parameter extractor of the block-xor PB-CAM.

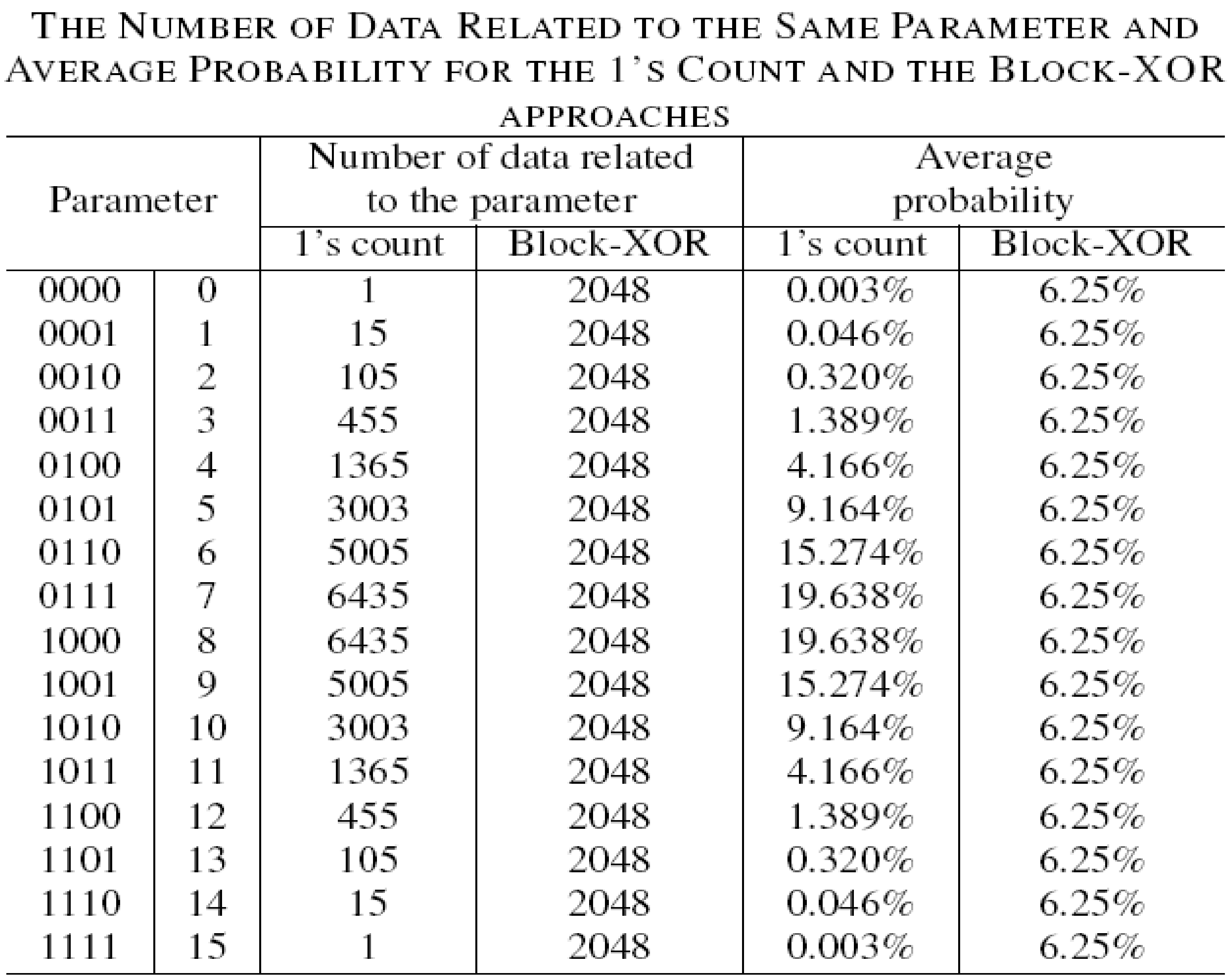
Mathematical Analysis for the Block-XOR PB-CAM

MATHEMATICAL FEATURE OF THE BLOCK-XOR APPROACH FOR 15-BIT RANDOM INPUT DATA

Parameter		Number of Data Related to the Same Parameter	Average Probability
0	0000	4×8×8×8=2048	6.25 %
1	0001	4×8×8×8=2048	6.25 %
2	0010	4×8×8×8=2048	6.25 %
3	0011	4×8×8×8=2048	6.25 %
4	0100	4×8×8×8=2048	6.25 %
5	0101	4×8×8×8=2048	6.25 %
6	0110	4×8×8×8=2048	6.25 %
7	0111	4×8×8×8=2048	6.25 %
8	1000	4×8×8×8=2048	6.25 %
9	1001	4×8×8×8=2048	6.25 %
10	1010	4×8×8×8=2048	6.25 %
11	1011	4×8×8×8=2048	6.25 %
12	1100	4×8×8×8=2048	6.25 %
13	1101	4×8×8×8=2048	6.25 %
14	1110	4×8×8×8=2048	6.25 %
15	1111	4×8×8×8=2048	6.25 %



Comparisons of two PB-CAMs



THE NUMBER OF DATA RELATED TO THE SAME PARAMETER AND AVERAGE PROBABILITY FOR THE 1'S COUNT AND THE BLOCK-XOR APPROACHES

Parameter		Number of data related to the parameter		Average probability	
		1's count	Block-XOR	1's count	Block-XOR
0000	0	1	2048	0.003%	6.25%
0001	1	15	2048	0.046%	6.25%
0010	2	105	2048	0.320%	6.25%
0011	3	455	2048	1.389%	6.25%
0100	4	1365	2048	4.166%	6.25%
0101	5	3003	2048	9.164%	6.25%
0110	6	5005	2048	15.274%	6.25%
0111	7	6435	2048	19.638%	6.25%
1000	8	6435	2048	19.638%	6.25%
1001	9	5005	2048	15.274%	6.25%
1010	10	3003	2048	9.164%	6.25%
1011	11	1365	2048	4.166%	6.25%
1100	12	455	2048	1.389%	6.25%
1101	13	105	2048	0.320%	6.25%
1110	14	15	2048	0.046%	6.25%
1111	15	1	2048	0.003%	6.25%

Previous Work & Observation

- The probability viewpoint proves that the block-xor approach reduce the number of comparison operations in 88% of the cases for 15-bit random input data compared with that of the 1’s count approach.

- However, for most of applications, their data distribution characteristics are specific rather than random.

Proposed Approach

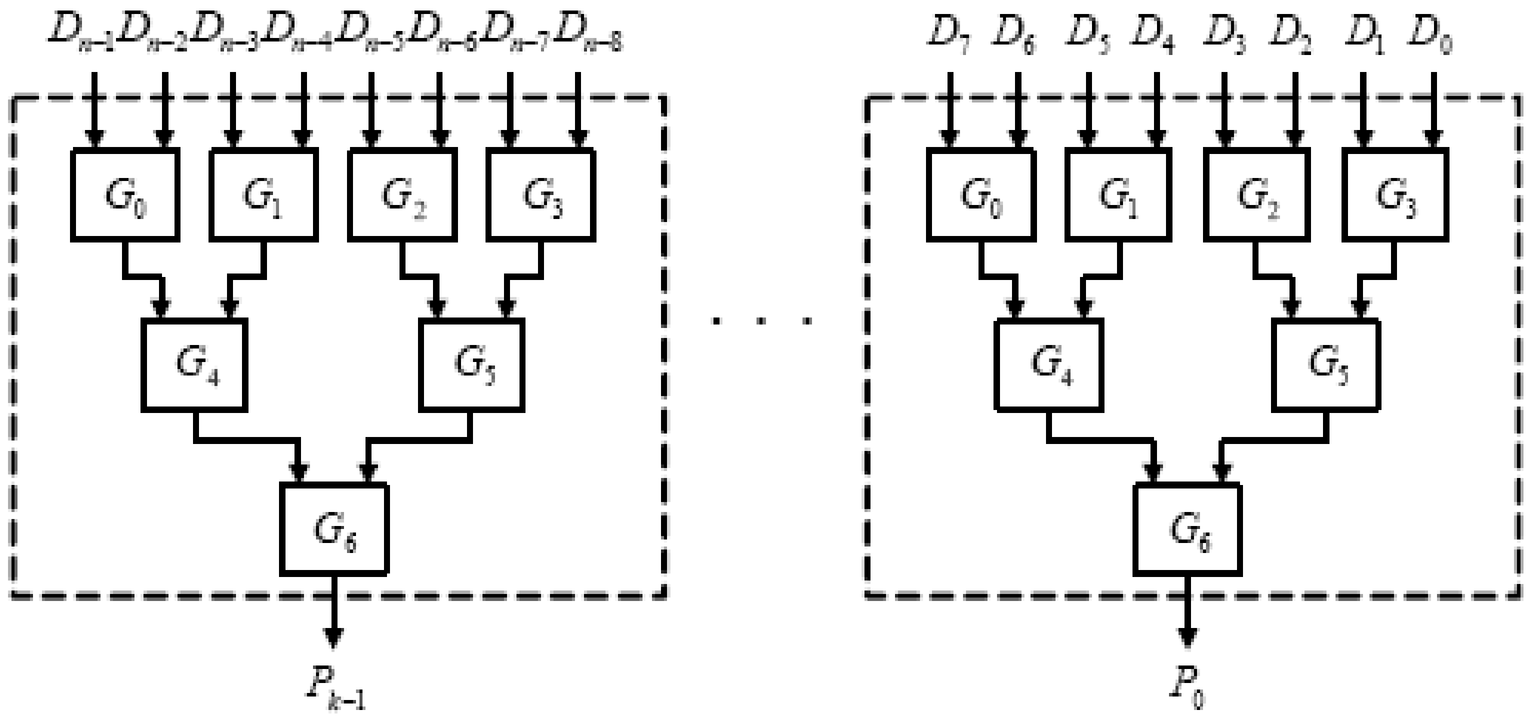

D_{n-1} D_{n-2} D_{n-3} D_{n-4} D_{n-5} D_{n-6} D_{n-7} D_{n-8}
G_0
G_1
G_2
G_3
G_4
G_5
G_6
P_{k-1}
. . . .
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
G_0
G_1
G_2
G_3
G_4
G_5
G_6
P_0

Fig. 4. n-bit block diagram of the proposed parameter extractor architecture.

Proposed Approach

- Suppose that we use basic logic gates (AND, OR, XOR, NAND, NOR, and NXOR) to synthesize a parameter extractor for a specific data type, which has 6^7^(n/8) different logic combinations.

- Obviously, the optimal combination of this parameter extractor can not be found in polynomial time.

Proposed Approach

- To synthesize a proper parameter extractor in polynomial time for a specific data type, we propose a gate-block selection algorithm to find an approximately optimal combination.

- We illustrate how to select proper logic gates to synthesize a parameter extractor for specific data type from mathematical analysis below.

Mathematical Analysis

- For a 2-input logic gate, let *p* be the probability of the output signal Y that is *one* state.
- The probability mass function of the output signal *Y* is given by

$$P_Y(y) = \begin{cases} 1-p & y=0, \\ p & y=1, \\ 0 & \text{otherwise.} \end{cases}$$

Mathematical Analysis

- Assume that the inputs are independent, if we use any 2-input logic gate as a parameter extractor to generate the parameter for 2-bit data, then the PB-CAM requires the average number of comparison operations in each data comparison process can be formulated as

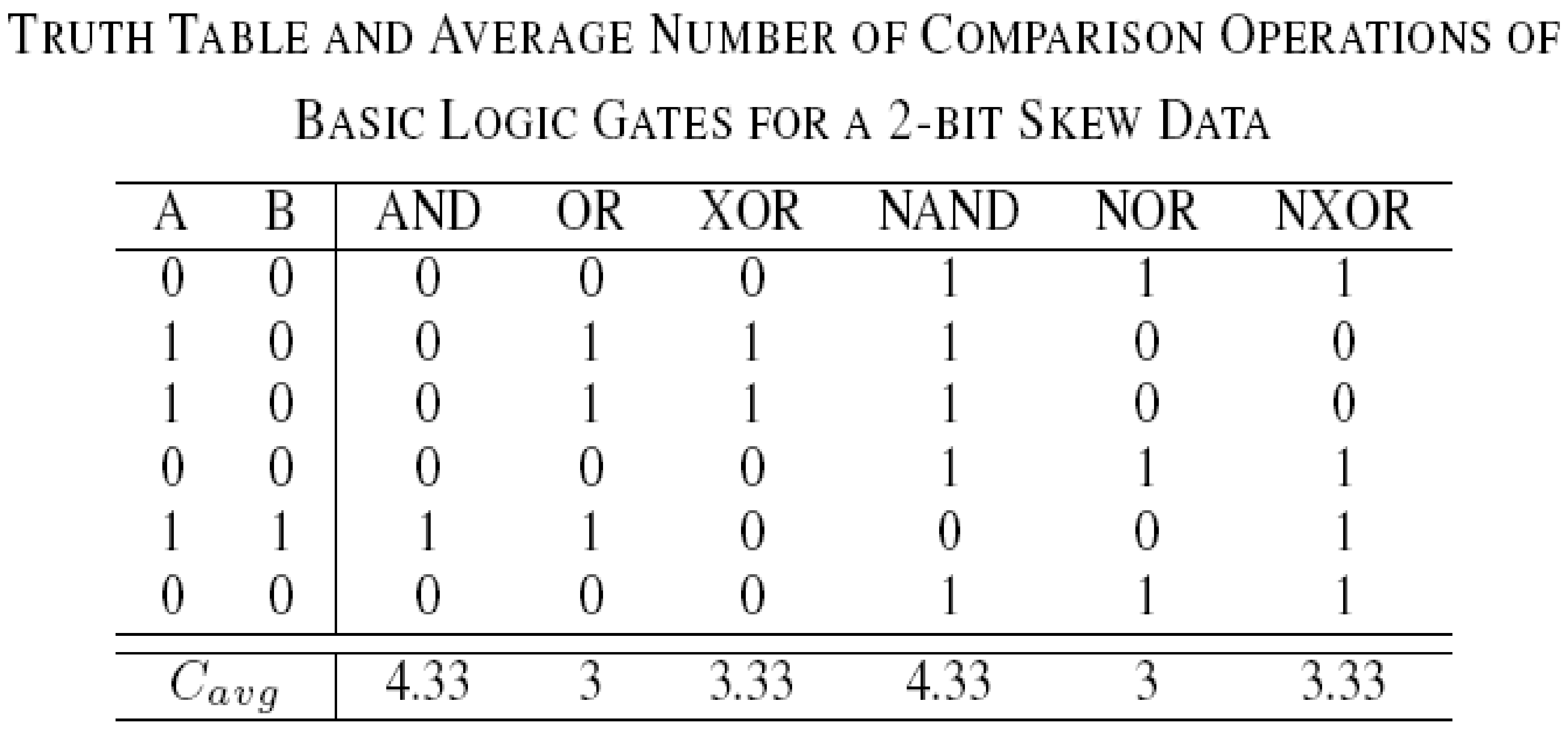
$$C_{avg} = N_0(1-p) + N_1 \cdot p$$

$$= N_0(\frac{N_0}{N_0+N_1}) + N_1(\frac{N_1}{N_0+N_1})$$

$$= \frac{{N_0}^2 + {N_1}^2}{N_0+N_1}$$

- where *N0* is the number of *zero* entries, and *N1* is the number of *one* entries for the generated parameters.

Mathematical Analysis



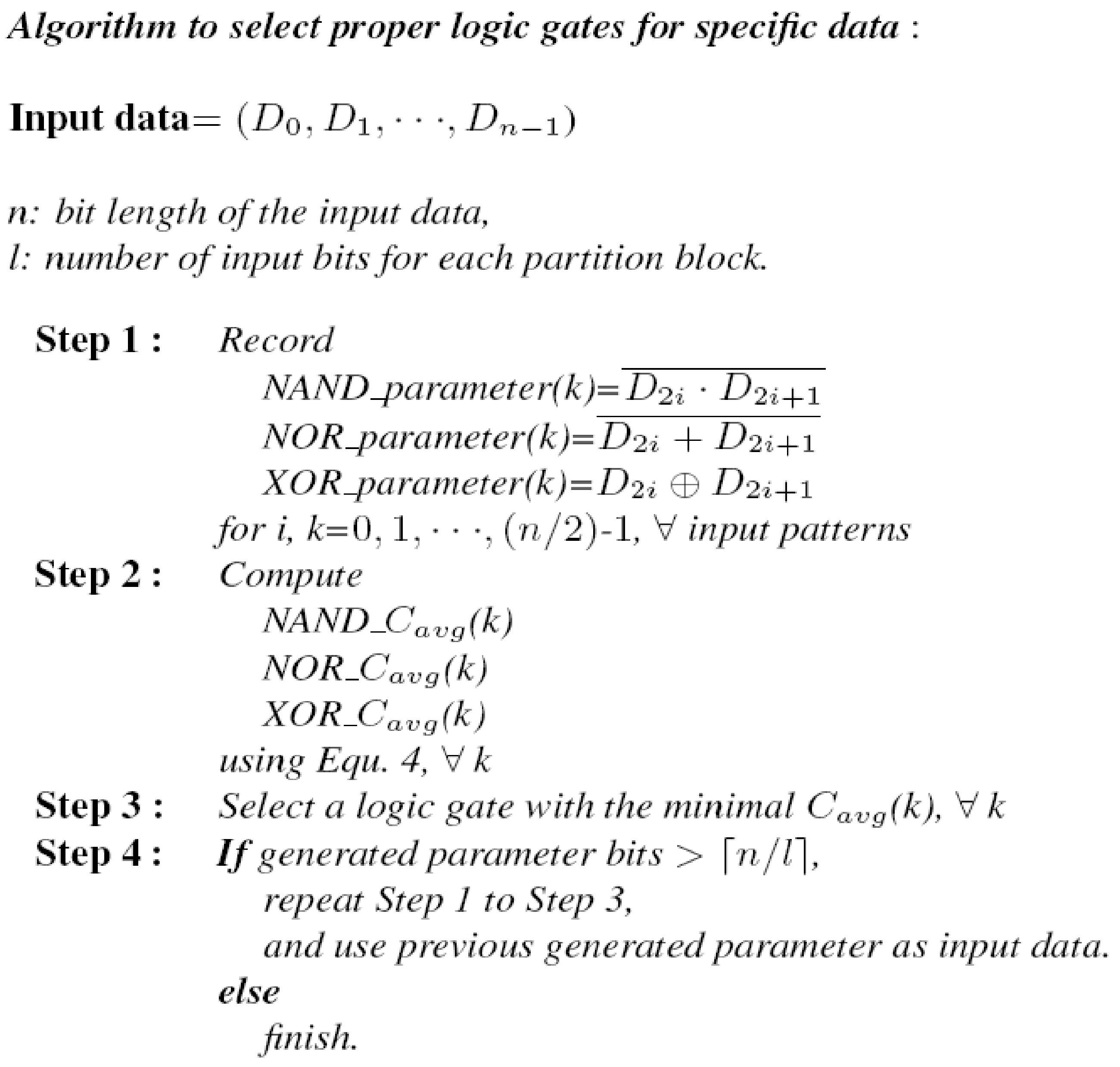
TRUTH TABLE AND AVERAGE NUMBER OF COMPARISON OPERATIONS OF BASIC LOGIC GATES FOR A 2-BIT SKEW DATA

A	B	AND	OR	XOR	NAND	NOR	NXOR
0	0	0	0	0	1	1	1
1	0	0	1	1	1	0	0
1	0	0	1	1	1	0	0
0	0	0	0	0	1	1	1
1	1	1	1	0	0	0	1
0	0	0	0	0	1	1	1
C_{avg}		4.33	3	3.33	4.33	3	3.33

Proposed Approach

- To reduce the complexity of our proposed algorithm and enhances the performance of the parameter extractor, our proposed approach only selects NAND, NOR, and XOR gates to synthesize the parameter extractor for our implementation.
- This is because that NAND and NOR are better than AND and OR in terms of the area, power, and speed.

Gate-Block Selection Algorithm



Algorithm to select proper logic gates for specific data :

Input data$= (D_0, D_1, \cdots, D_{n-1})$

n: bit length of the input data,
l: number of input bits for each partition block.

```
Step 1 :  Record
              $NAND\_parameter(k) = \overline{D_{2i} \cdot D_{2i+1}}$
              $NOR\_parameter(k) = \overline{D_{2i} + D_{2i+1}}$
              $XOR\_parameter(k) = D_{2i} \oplus D_{2i+1}$
          for $i, k = 0, 1, \cdots, (n/2)$-1, $\forall$ input patterns
Step 2 :  Compute
              $NAND\_C_{avg}(k)$
              $NOR\_C_{avg}(k)$
              $XOR\_C_{avg}(k)$
          using Equ. 4, $\forall k$
Step 3 :  Select a logic gate with the minimal $C_{avg}(k)$, $\forall k$
Step 4 :  If generated parameter bits $> \lceil n/l \rceil$,
              repeat Step 1 to Step 3,
              and use previous generated parameter as input data.
          else
              finish.
```

Fig. 5. Gate-Block Selection Algorithm.

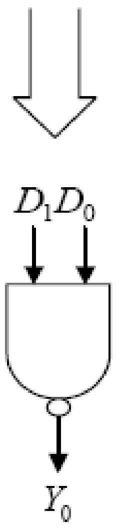
An Example

D_3	D_2	D_1	D_0
0	0	1	1
1	0	0	1
0	0	0	0
0	0	1	1
0	0	1	1
0	1	0	1
0	0	1	0
1	1	1	1
1	0	0	1
0	0	1	1
0	0	1	0
1	0	0	1
0	0	0	0
0	1	1	1
1	1	1	1
0	0	1	1

D_1D_0	NAND	NOR	XOR
1 1	0	0	0
0 1	1	0	1
0 0	1	1	0
1 1	0	0	0
1 1	0	0	0
0 1	1	0	1
1 0	1	0	1
1 1	0	0	0
0 1	1	0	1
1 1	0	0	0
1 0	1	0	1
0 1	1	0	1
0 0	1	1	0
1 1	0	0	0
1 1	0	0	0
1 1	0	0	0

D_1D_0	C_{avg}
NAND	8
NOR	12.5
XOR	8.5

minimal



(a)

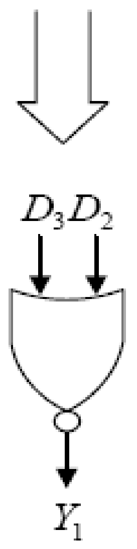
An Example

D_3	D_2	D_1	D_0
0	0	1	1
1	0	0	1
0	0	0	0
0	0	1	1
0	0	1	1
0	1	0	1
0	0	1	0
1	1	1	1
1	0	0	1
0	0	1	1
0	0	1	0
1	0	0	1
0	0	0	0
0	1	1	1
1	1	1	1
0	0	1	1

D_3	D_2	NAND	NOR	XOR
0	0	1	1	0
1	0	1	0	1
0	0	1	1	0
0	0	1	1	0
0	0	1	1	0
0	1	1	0	1
0	0	1	1	0
1	1	0	0	0
1	0	1	0	1
0	0	1	1	0
0	0	1	1	0
1	0	1	0	1
0	0	1	1	0
0	1	1	0	1
1	1	0	0	0
0	0	1	1	0

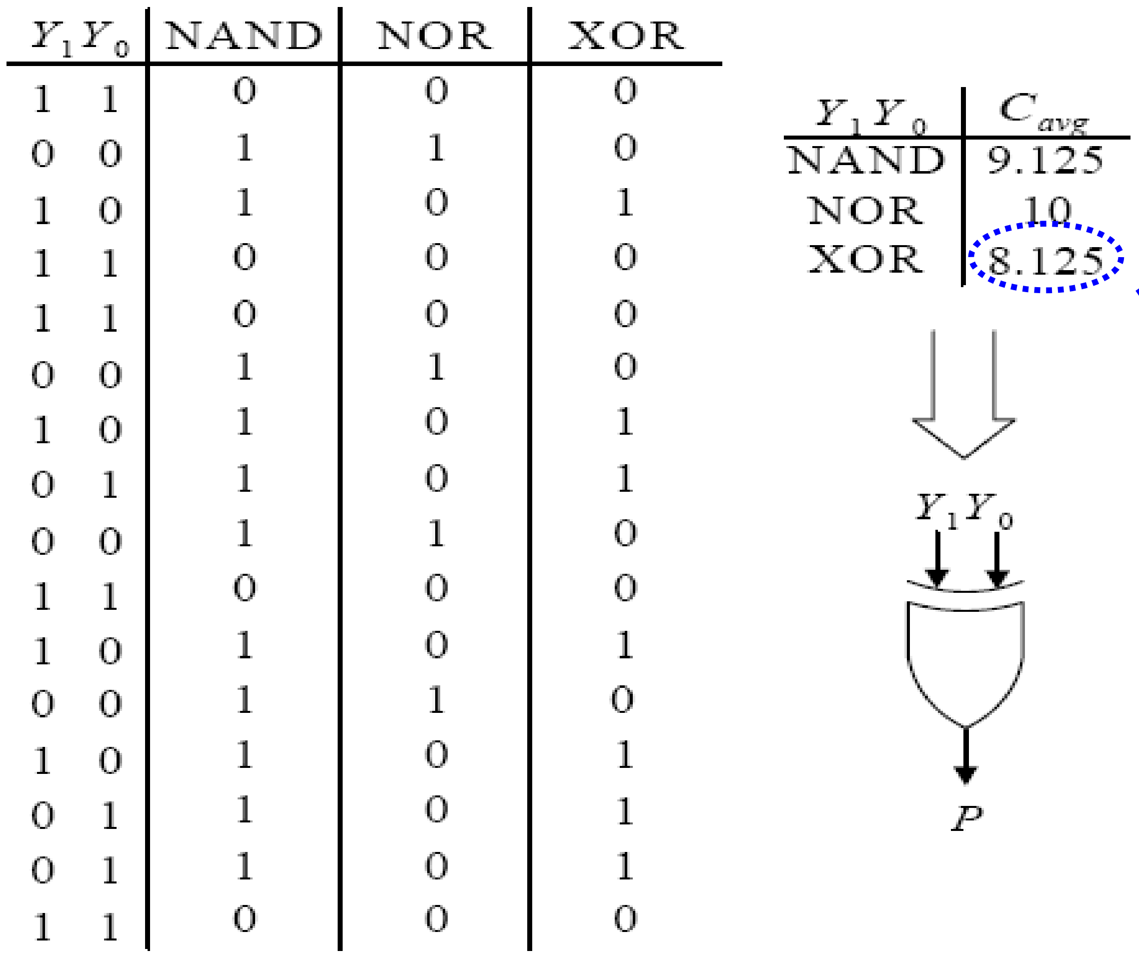
D_3D_2	C_{avg}
NAND	12.5
NOR	8.125
XOR	9.125

minimal



(b)

An Example



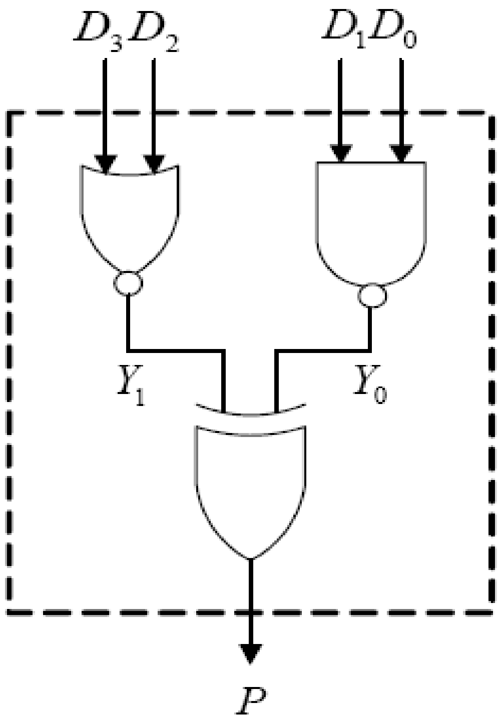
$Y_1 Y_0$	NAND	NOR	XOR
1 1	0	0	0
0 0	1	1	0
1 0	1	0	1
1 1	0	0	0
1 1	0	0	0
0 0	1	1	0
1 0	1	0	1
0 1	1	0	1
0 0	1	1	0
1 1	0	0	0
1 0	1	0	1
0 0	1	1	0
1 0	1	0	1
0 1	1	0	1
0 1	1	0	1
1 1	0	0	0

$Y_1 Y_0$	C_{avg}
NAND	9.125
NOR	10
XOR	8.125

minimal

(c)

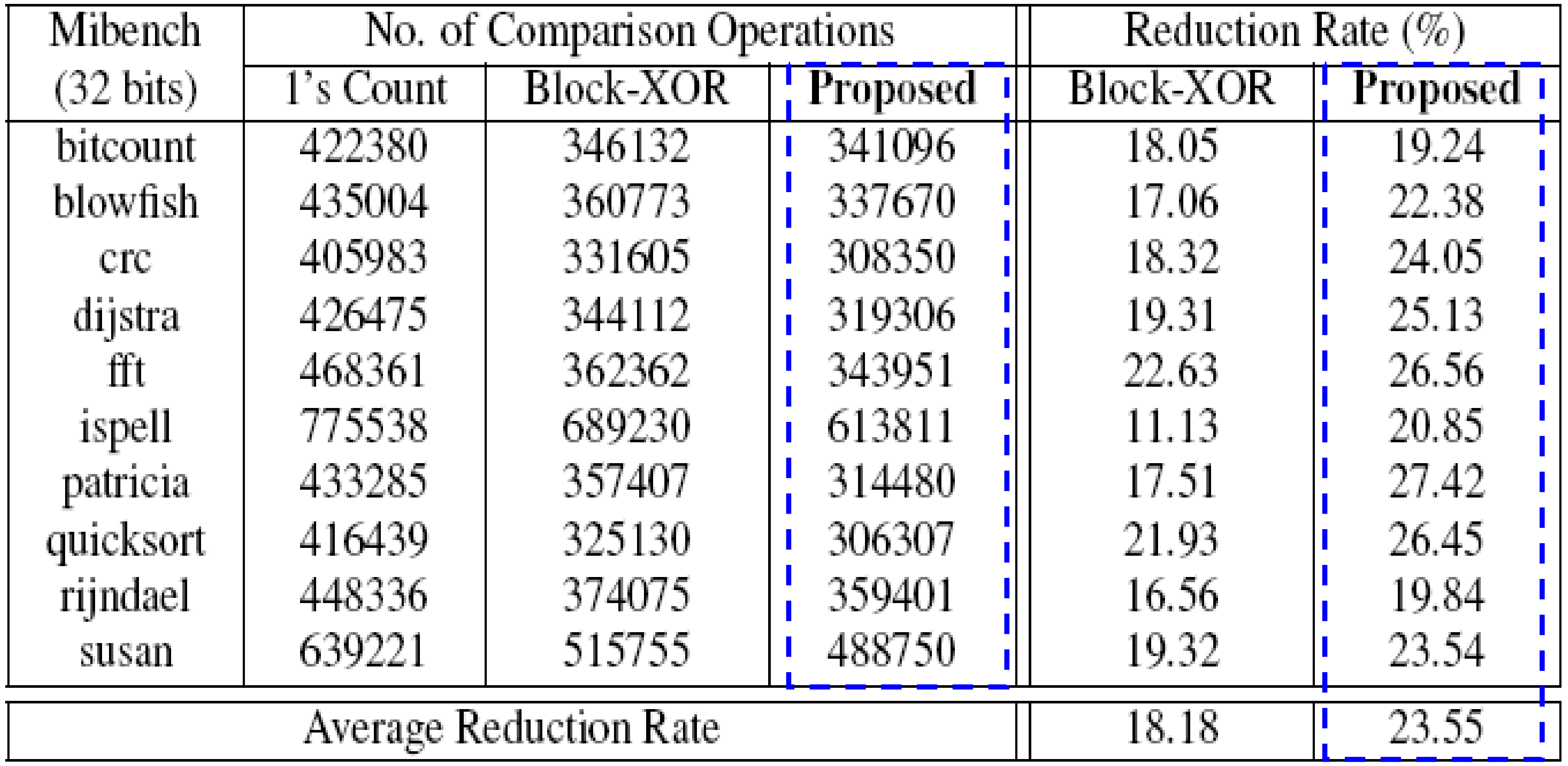
An Example



(d)

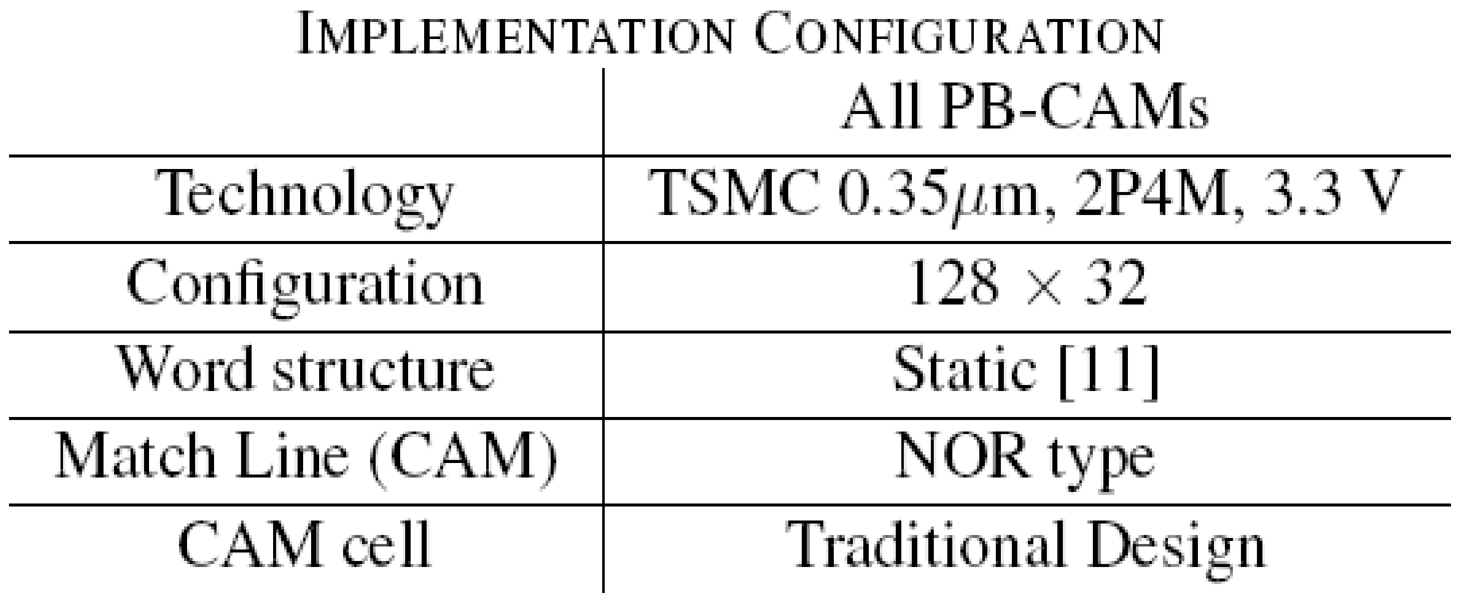
Experimental Results

NUMBER OF COMPARISON OPERATIONS FOR DIFFERENT PRE-COMPUTATION APPROACHES



Mibench (32 bits)	No. of Comparison Operations			Reduction Rate (%)	
	1's Count	Block-XOR	**Proposed**	Block-XOR	**Proposed**
bitcount	422380	346132	341096	18.05	19.24
blowfish	435004	360773	337670	17.06	22.38
crc	405983	331605	308350	18.32	24.05
dijstra	426475	344112	319306	19.31	25.13
fft	468361	362362	343951	22.63	26.56
ispell	775538	689230	613811	11.13	20.85
patricia	433285	357407	314480	17.51	27.42
quicksort	416439	325130	306307	21.93	26.45
rijndael	448336	374075	359401	16.56	19.84
susan	639221	515755	488750	19.32	23.54
Average Reduction Rate				18.18	23.55

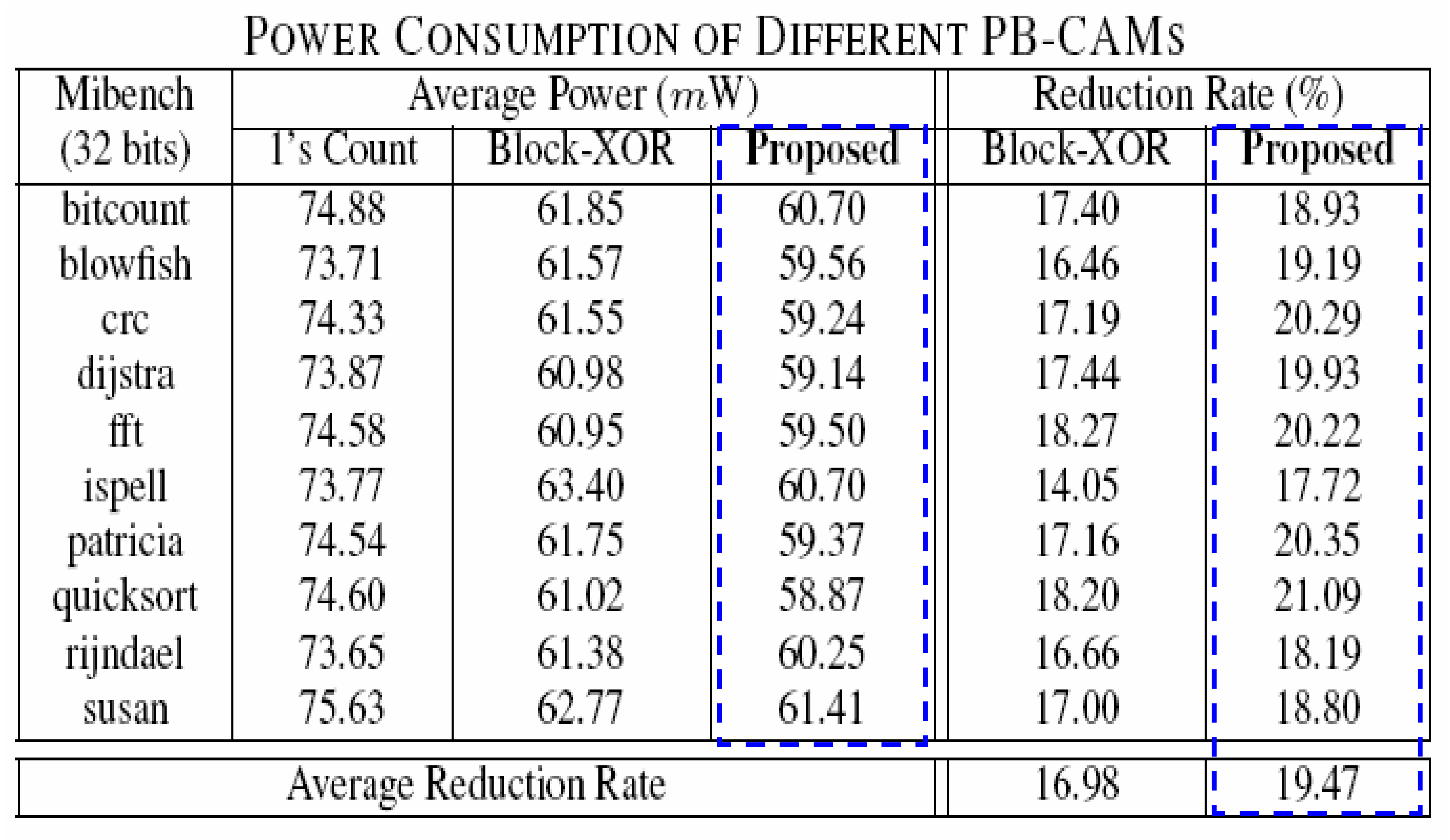
Experimental Results



IMPLEMENTATION CONFIGURATION

	All PB-CAMs
Technology	TSMC 0.35μm, 2P4M, 3.3 V
Configuration	128 $\times$ 32
Word structure	Static [11]
Match Line (CAM)	NOR type
CAM cell	Traditional Design

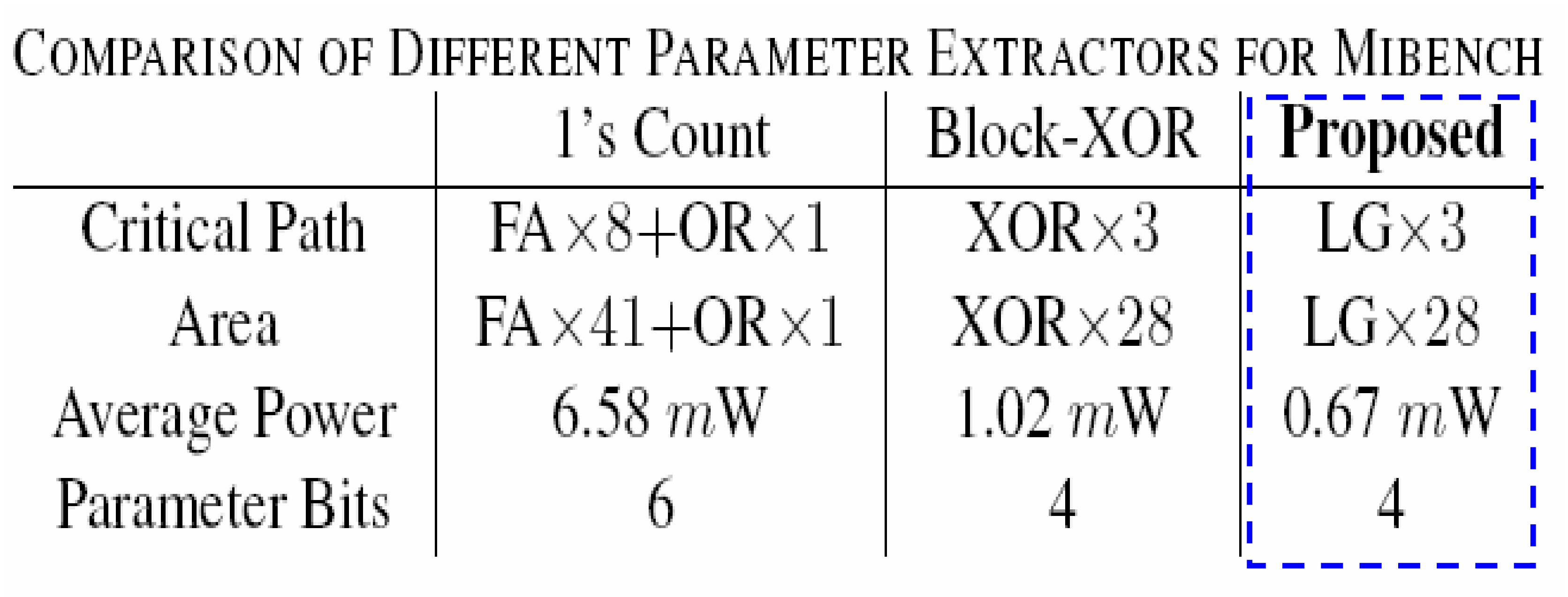
Experimental Results



POWER CONSUMPTION OF DIFFERENT PB-CAMS

Mibench (32 bits)	Average Power (mW)			Reduction Rate (%)	
	1's Count	Block-XOR	**Proposed**	Block-XOR	**Proposed**
bitcount	74.88	61.85	60.70	17.40	18.93
blowfish	73.71	61.57	59.56	16.46	19.19
crc	74.33	61.55	59.24	17.19	20.29
dijstra	73.87	60.98	59.14	17.44	19.93
fft	74.58	60.95	59.50	18.27	20.22
ispell	73.77	63.40	60.70	14.05	17.72
patricia	74.54	61.75	59.37	17.16	20.35
quicksort	74.60	61.02	58.87	18.20	21.09
rijndael	73.65	61.38	60.25	16.66	18.19
susan	75.63	62.77	61.41	17.00	18.80
Average Reduction Rate				16.98	19.47

Experimental Results


COMPARISON OF DIFFERENT PARAMETER EXTRACTORS FOR MIBENCH

	1's Count	Block-XOR	**Proposed**
Critical Path	FA$\times$8+OR$\times$1	XOR$\times$3	LG$\times$3
Area	FA$\times$41+OR$\times$1	XOR$\times$28	LG$\times$28
Average Power	6.58 mW	1.02 mW	0.67 mW
Parameter Bits	6	4	4

LG : Logic Gate (which is NAND, NOR, or XOR)

FA : Full Adder

Conclusion

- In this paper, a gate-block selection algorithm has been proposed, which can synthesize a proper parameter extractor of the PB-CAM for a specific data type.
- As shown in our experimental results, the proposed PB-CAM is very suitable for specific applications such as embedded systems.

Thank You !