Index Generation Functions: Logic Synthesis for Pattern Matching

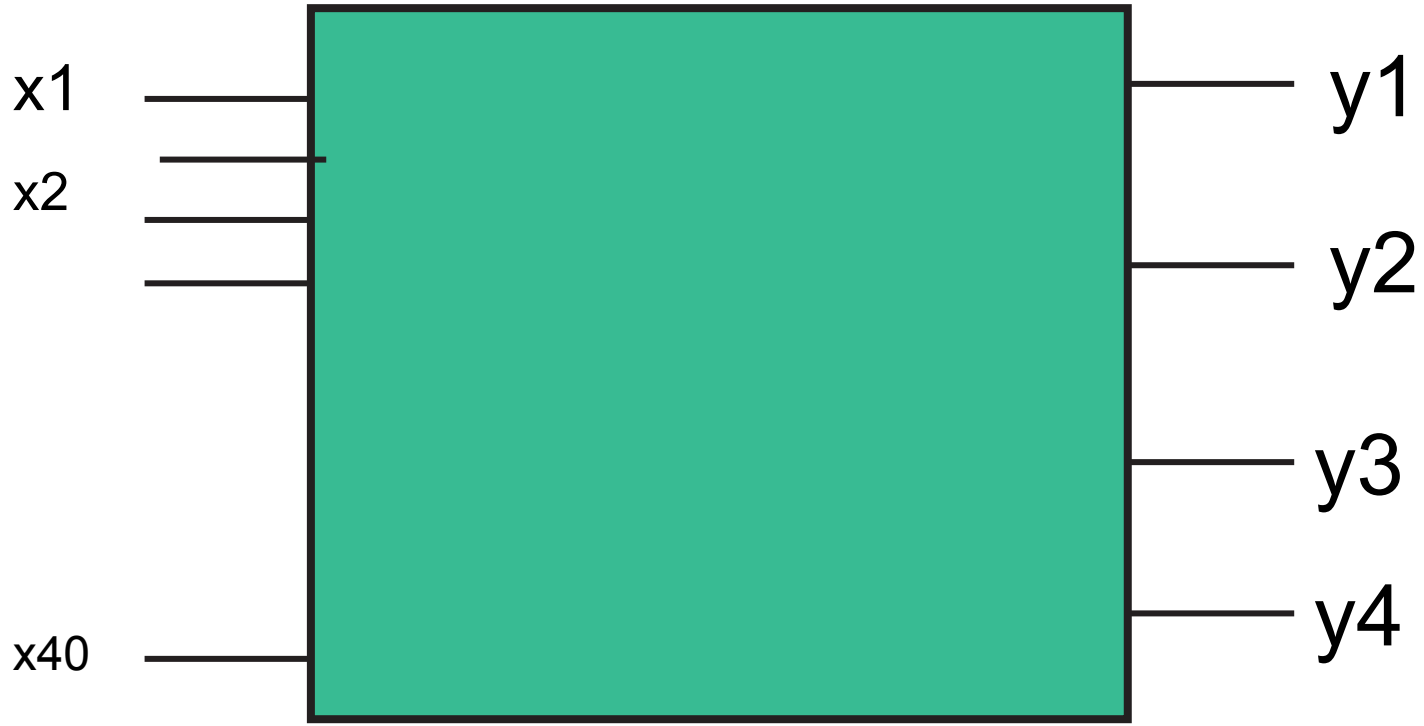
Tsutomu Sasao

Meiji University, Kanagawa, Japan

Q: We have 10 vectors of 40 bits. Design a circuit for pattern matching.

0110000100010000101001100001000100001010
0101111101101010001101011111011010100011
1111010101110111000011110101011101110001
0001111000010001011100011110000100010111
0011111000000100010100111100000001000101
0111001001000100100101110010010001001001
0010001110001111001000100011100011110010
1111111111010001111000100011100011110010
1110111000110001011011101110001100010110
1010000110100100001110100001101001000011

Circuit



Methods

1. Use a look up table.
2. Convert the table into Verilog code and use logic synthesis tool to generate an FPGA.
3. Use Xilinx CAM IP core.
4. Use a new method.

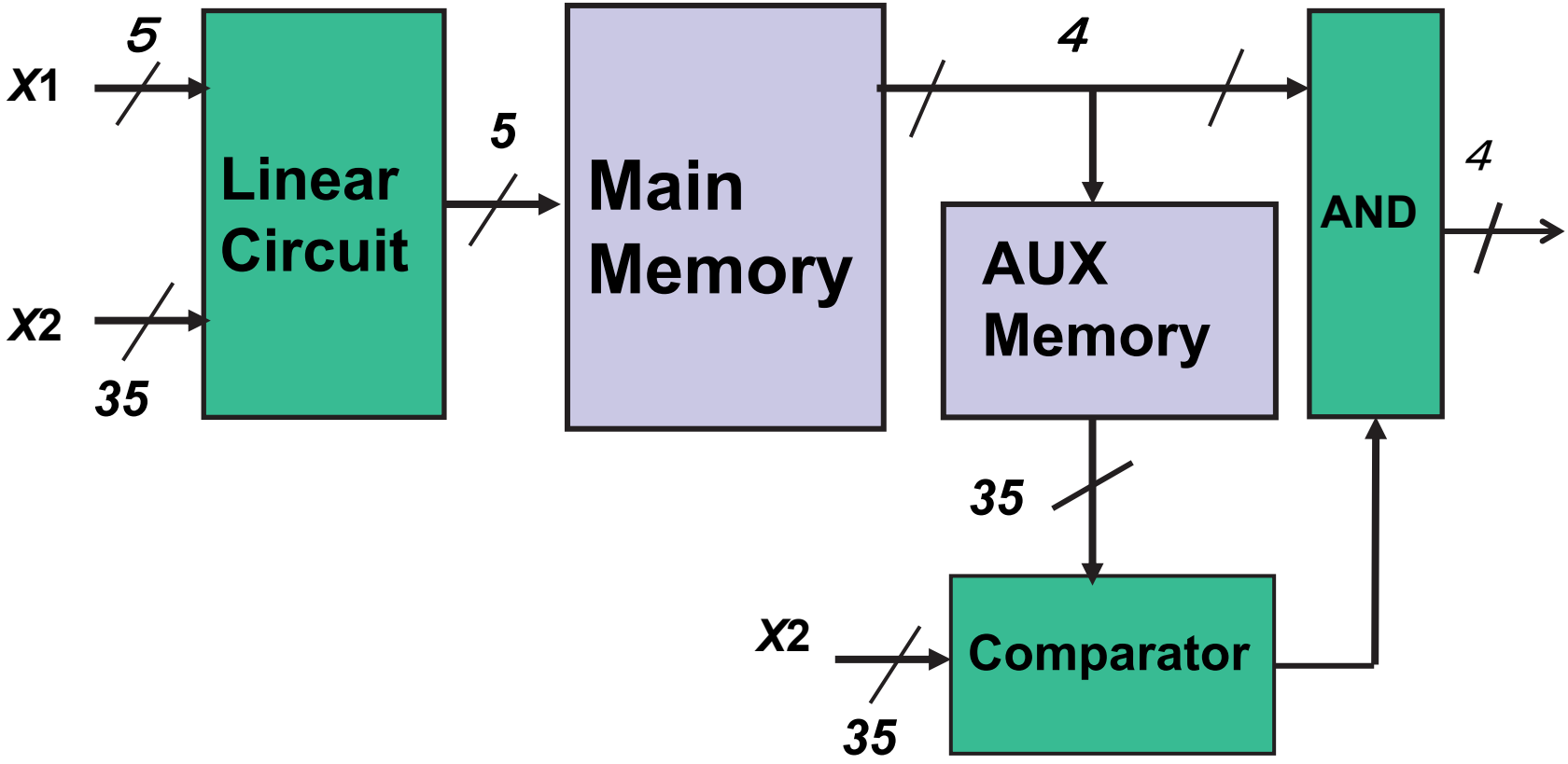
New Method

1. To distinguish 10 vectors, 5 bits are sufficient, in most cases.
2. Find 5 bits to distinguish 10 vectors.
3. Given an input vector, check if there exist a vector that matches the 5 bits.
4. If such a vector exists, check if other bits of the vector are equal to the input.

We have 10 vectors of 40 bits.
Design a circuit for pattern matching.

01100001000010000101001100001000100001010
01011111011010100011010111111011010100011
11110101011101110000111101010111011110001
00011110000100010111000111100001000010111
00111100000001000101001111000000010001001
01110010001001001011100100100010010010001
00100011100011110010001000111000111100010
11111111110100011110001000111000111100010
11101110001100010110111011100011000010110
10100001101001000011101000011010010000011

Index Generation Unit (IGU)



Incompletely Specified Index Generation Function

$$f : D \rightarrow \{1,2,...,k\},$$

$$\text{where } D \subseteq B^n, B = \{0,1\}.$$

Number of Variables to Represent Incompletely Specified Index Generation Functions

Index Generation Function

x1	x2	x3	x4	x5	x6	x7	index
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9
1	1	1	1	1	1	0	10

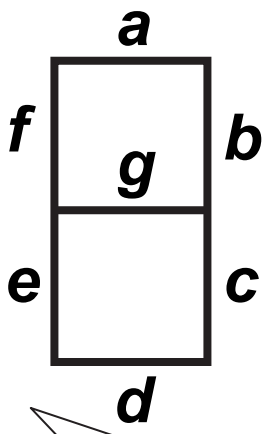
$n = 7$

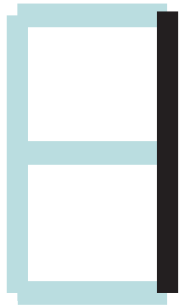
$k = 10$

Implement such functions by programmable devices

7-Segment to BCD Converter

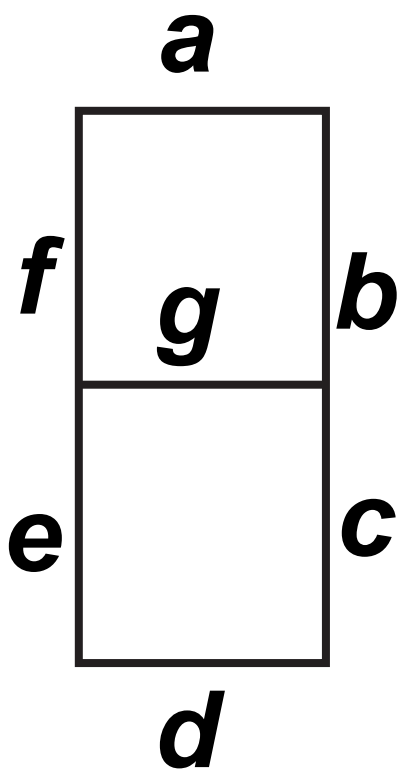
7segments							BCD
a	***b***	***c***	***d***	***e***	***f***	***g***	
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9
1	1	1	1	1	1	0	10

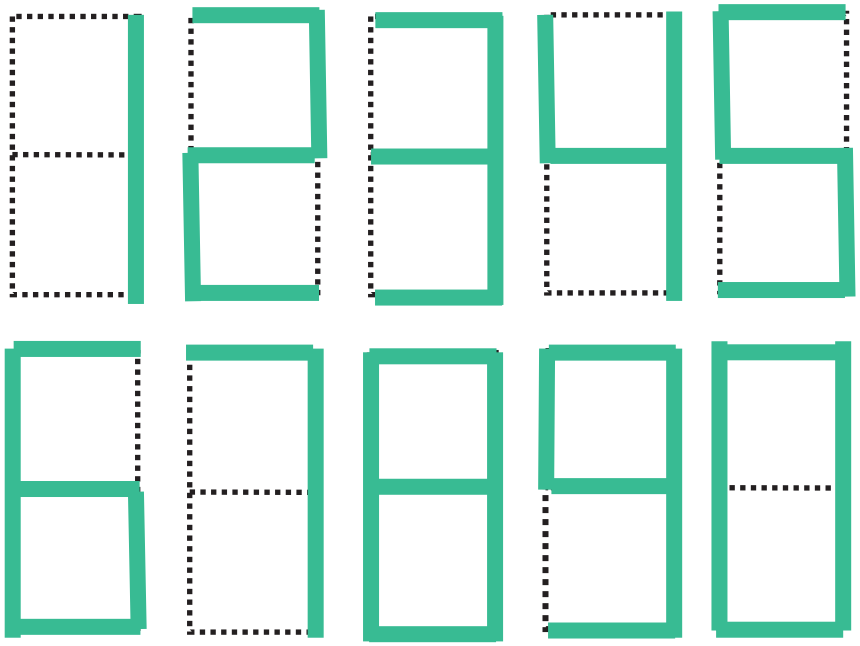




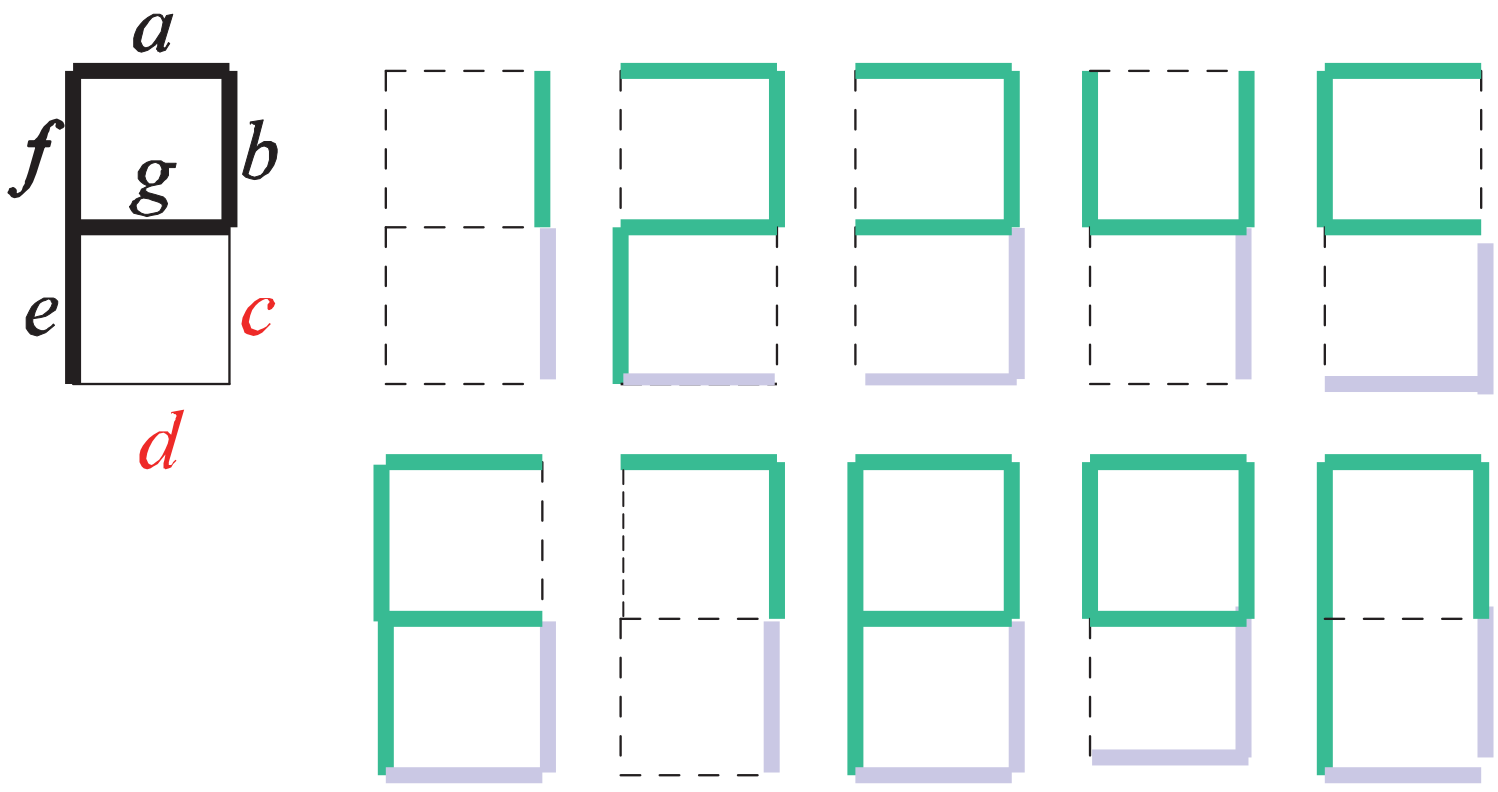
Do you need all the segments to identify the numbers?

Q: Which Segments are Necessary?



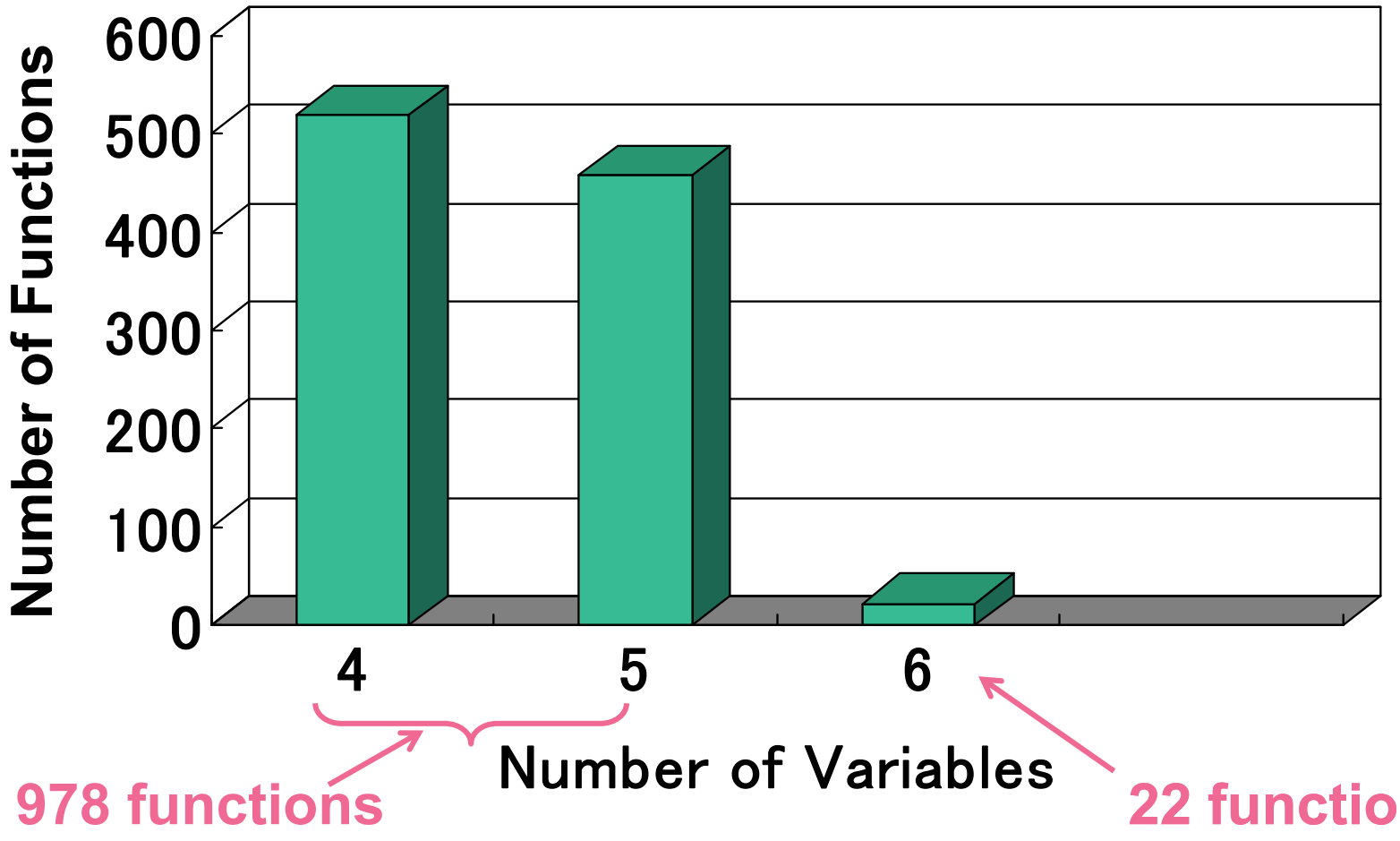


Five Segments are Necessary :
{a,b,e,f,g}



How many variables, on the average, are necessary to represent incompletely specified index generation functions with 7 variables and weight 10?

Distribution of functions that require 4, 5 and 6 variables for k=10, n=7



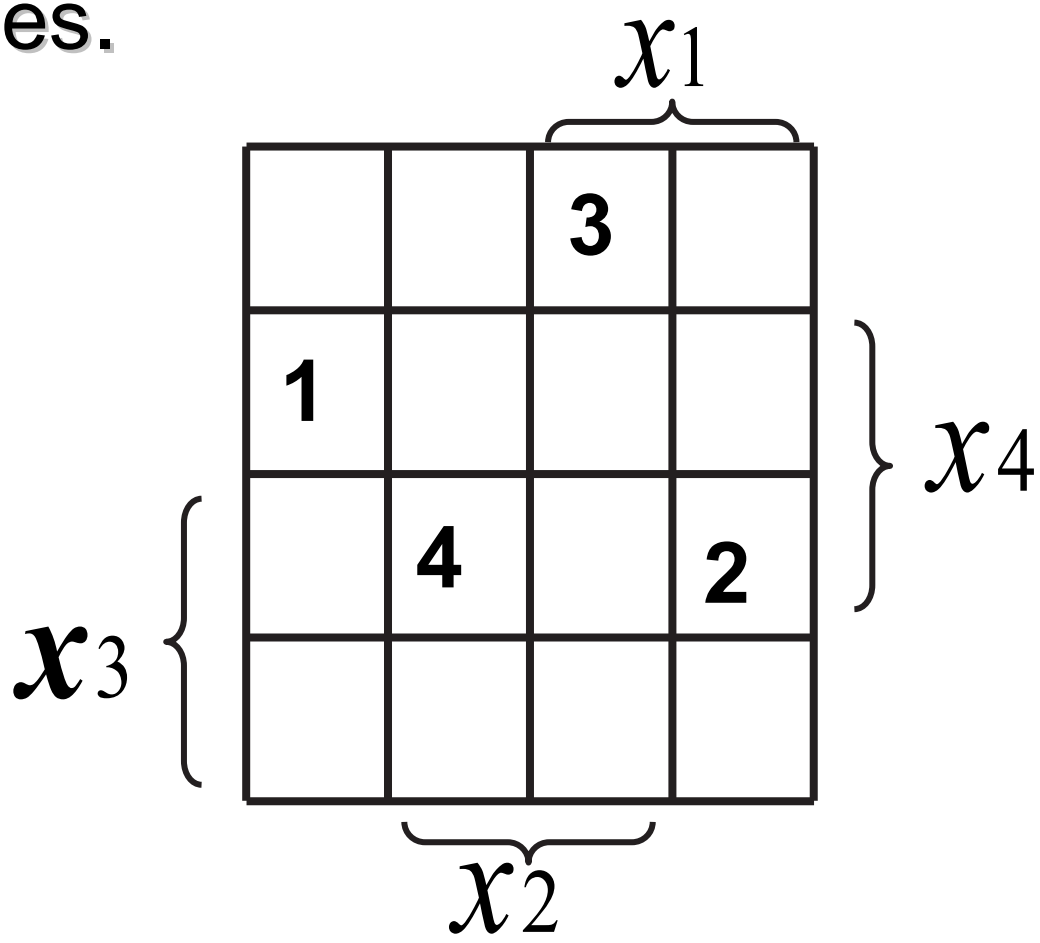
Property

To represent an incompletely specified index generation function with weight *k*, $2\lceil \log_2(k+1) \rceil - 3$ variables are sufficient, *f*or most cases, when $k \geq 7$.

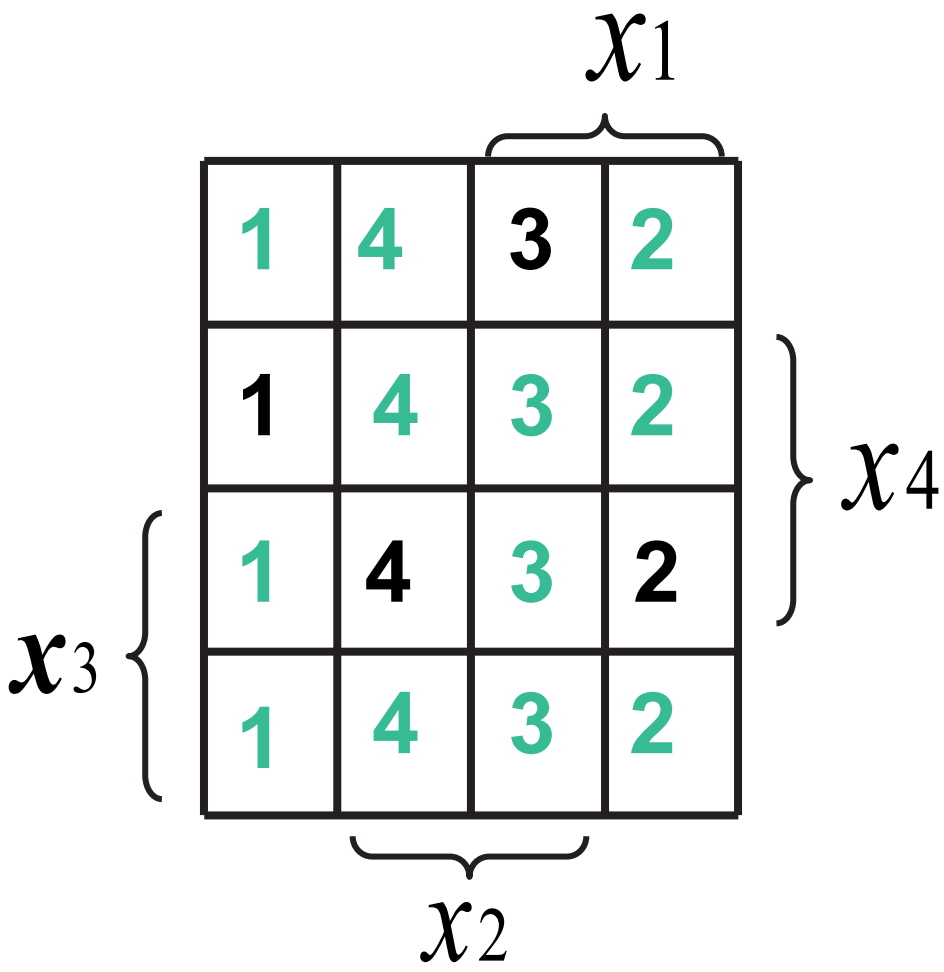
Example: When *k*=127.

$$2\lceil \log_2(k+1) \rceil - 3 = 2 \times 7 - 3 = 11$$

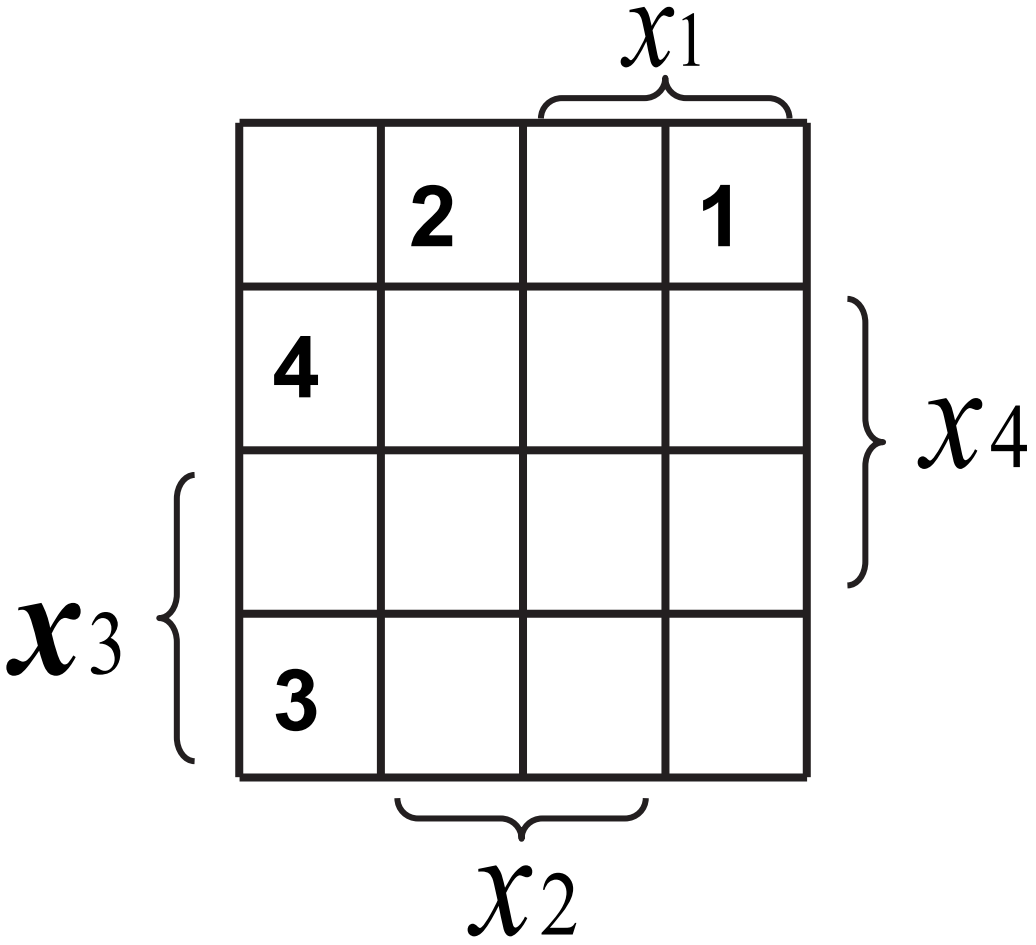
If each column has at most one non-zero element in a decomposition chart, then *f* can be represented with only the column variables.



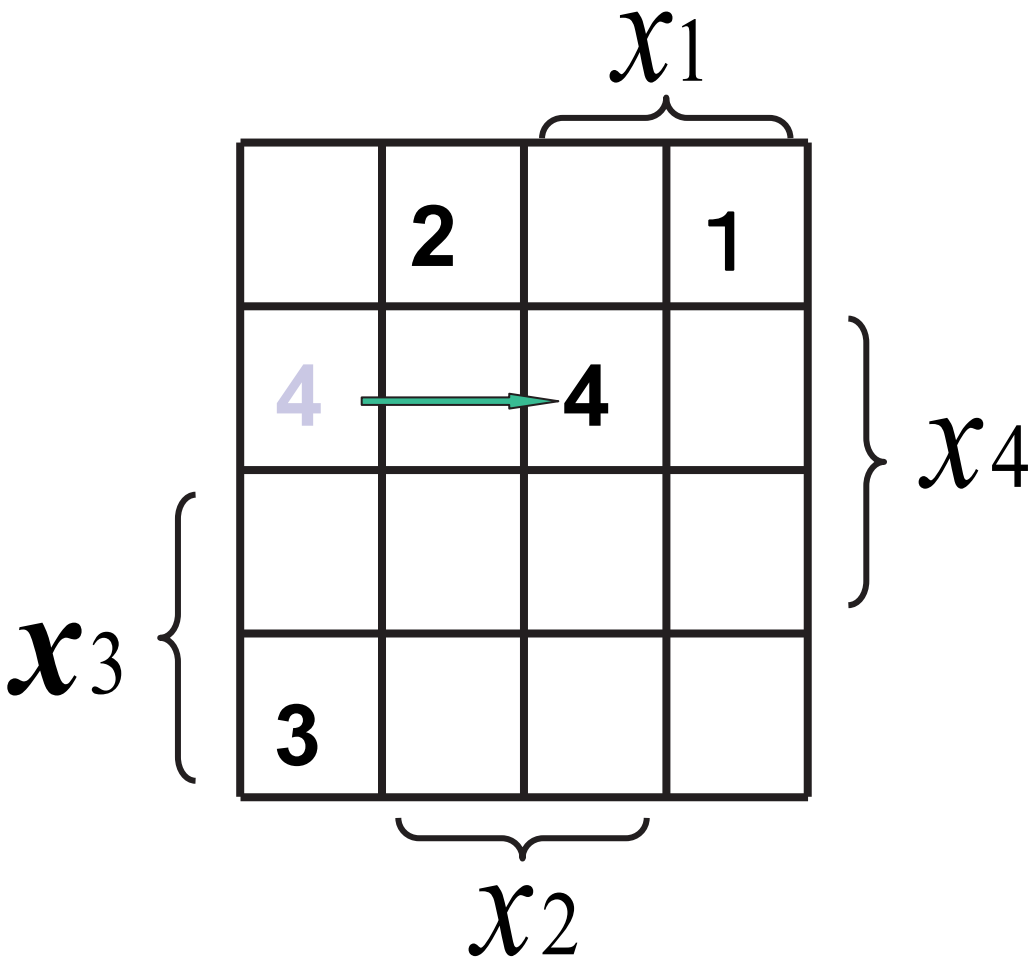
$$f = 1 \cdot \overline{x_1}\,\overline{x_2} \vee 4 \cdot \overline{x_1}\,x_2 \vee 3 \cdot x_1 x_2 \vee 2 \cdot x_1\,\overline{x_2}$$



Three variables are necessary to represent this function.



If the element 4 is moved right two squares, then *f* is represented by only x_1 and x_2



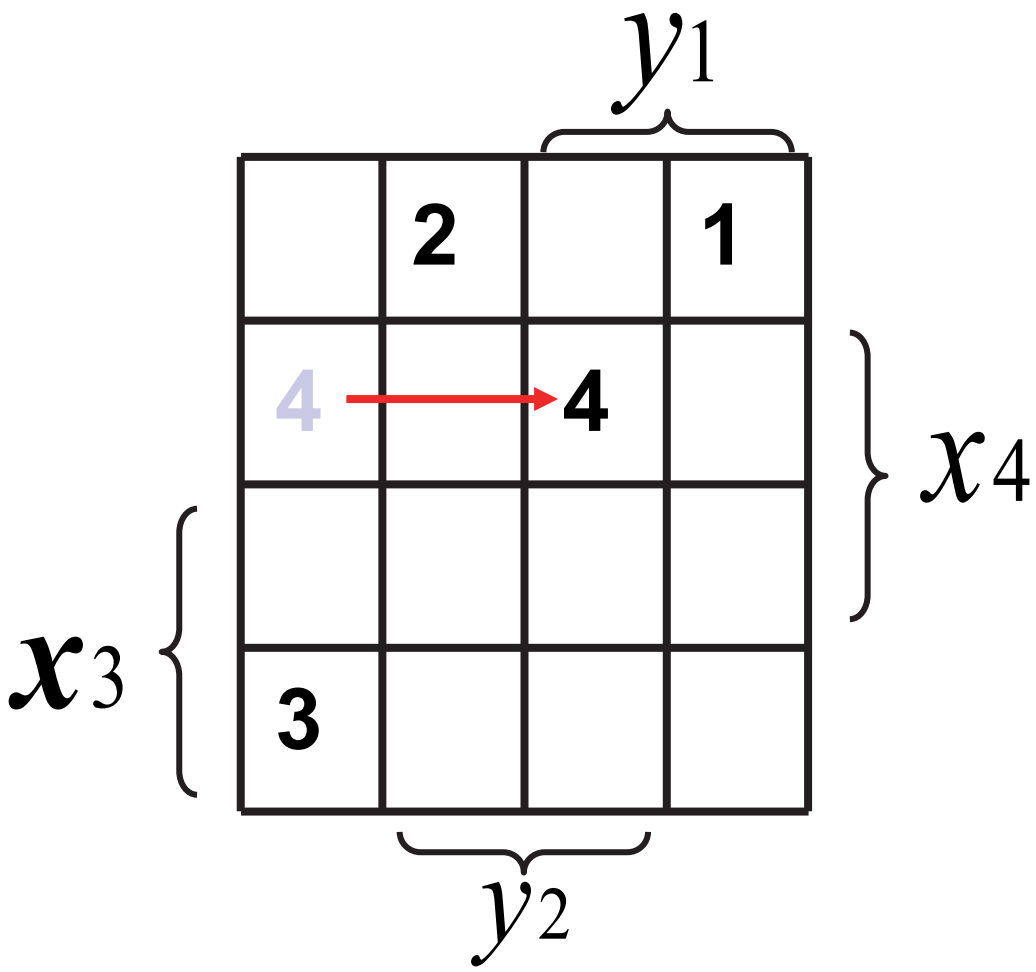
A linear transformation permutes elements.

$$y_1 = x_1 \oplus x_4$$

$$y_2 = x_2 \oplus x_4$$

$$y_3 = x_3$$

$$y_4 = x_4$$



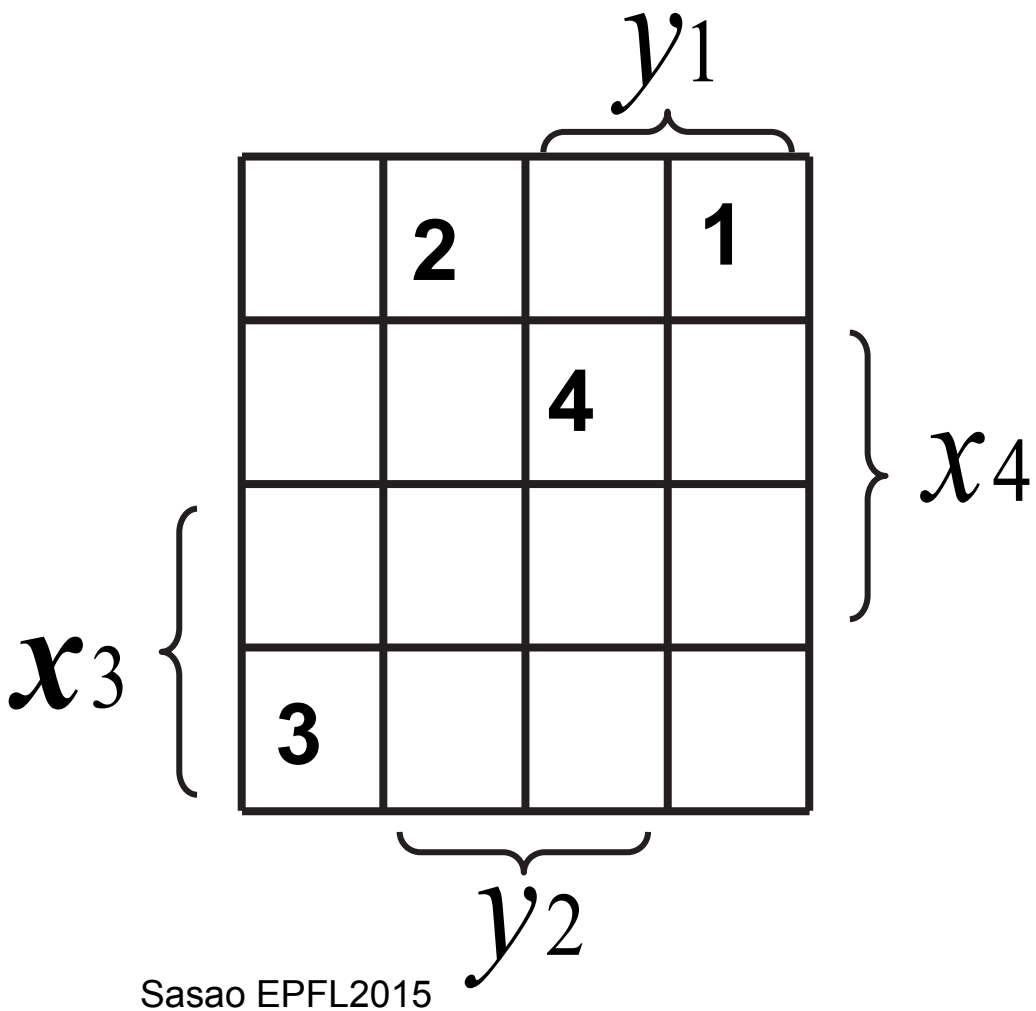
In this case, each column has at most one non-zero element.

$$y_1 = x_1 \oplus x_4$$

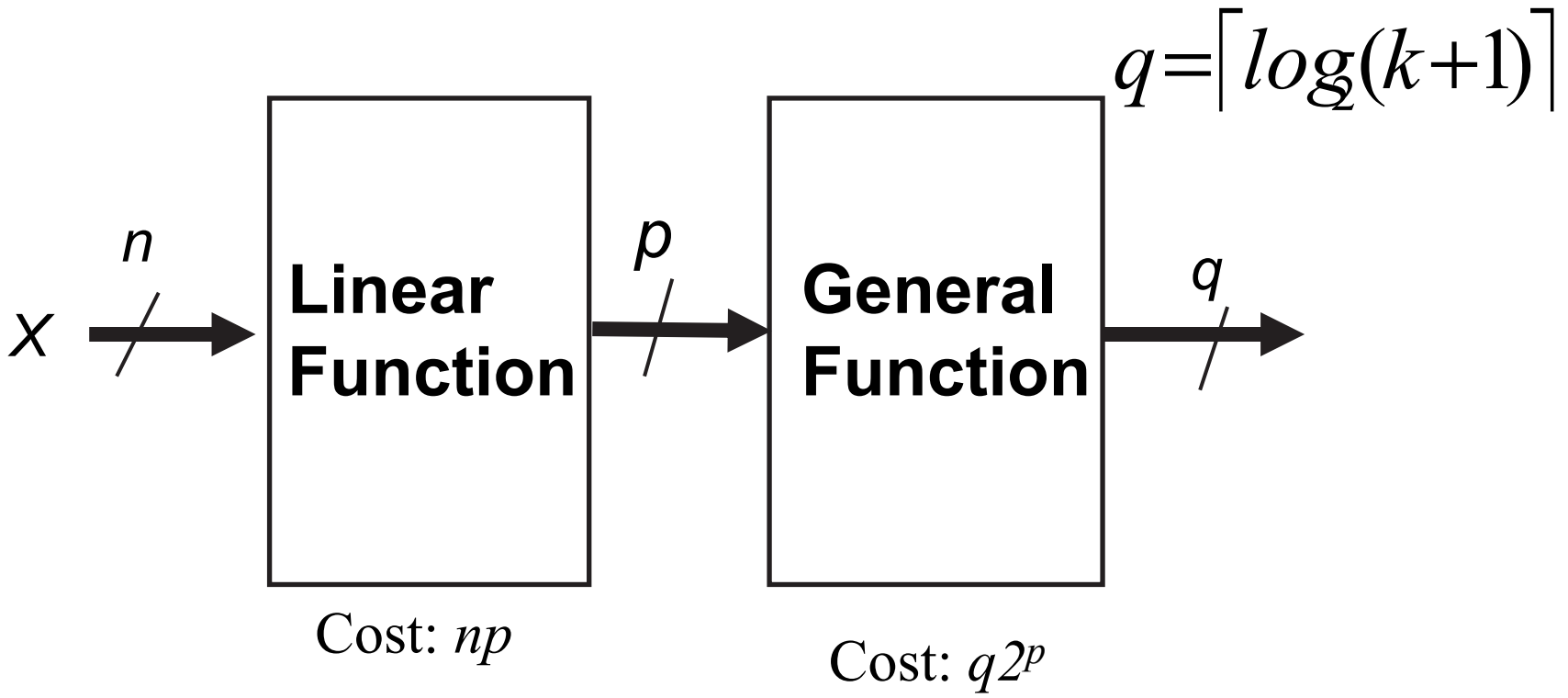
$$y_2 = x_2 \oplus x_4$$

$$y_3 = x_3$$

$$y_4 = x_4$$



Linear Decomposition



Compound Variables

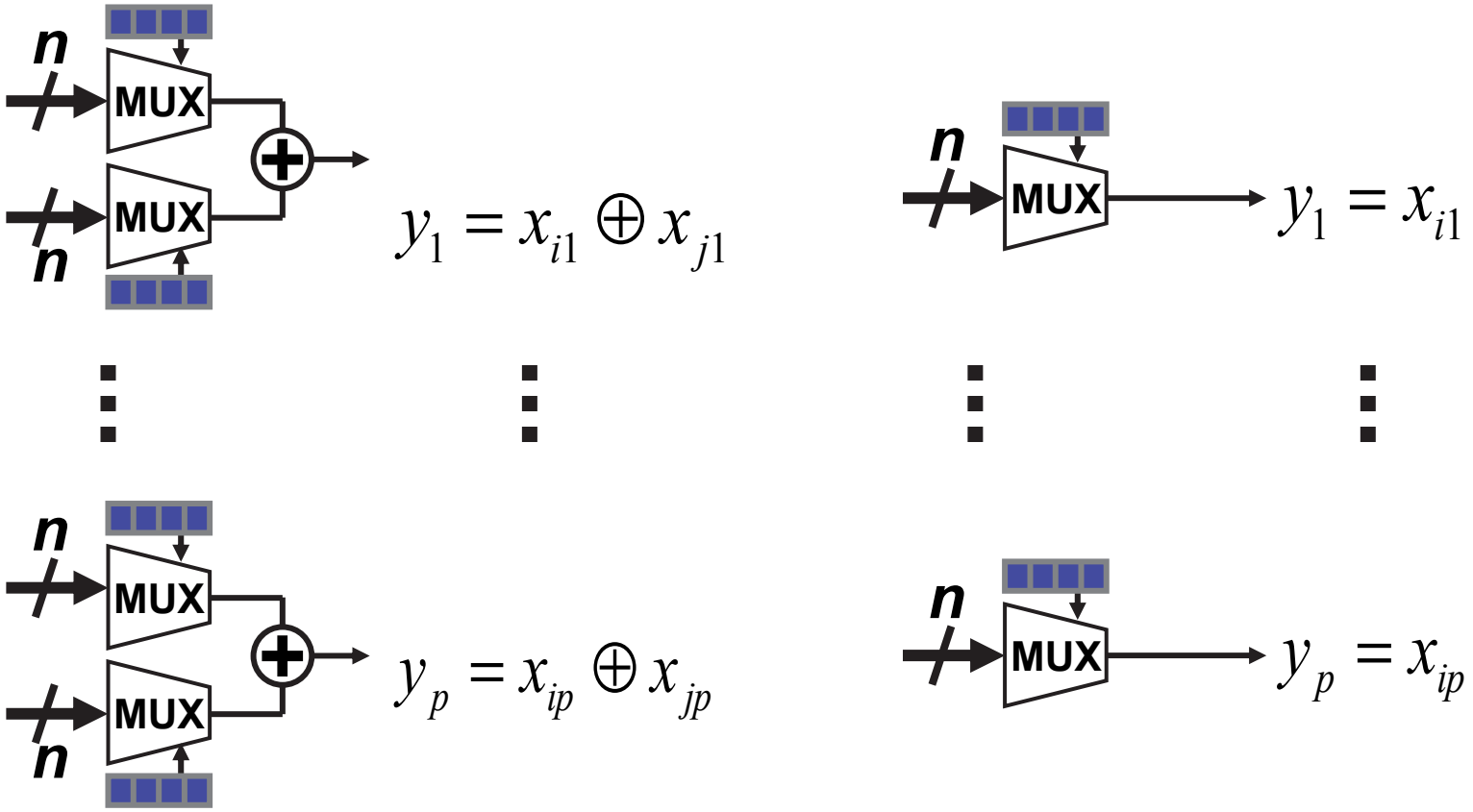
- General form:

$$y = a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n$$

$$\boldsymbol{a_i} \in \{0,1\}$$

- **Compound degree:** The number of coefficients with a_i=1 **.**
- A variable with the compound degree 1 is **primitive**.

Linear Circuits



Compound degree 2

Compound degree 1

1-out-of-7 code to Index Converter

1-out-of-7 code *x*7 *x*6*x*5 *x*4*x*3*x*2*x*1	Index
0 0 0 0 0 0 1	1
0 0 0 0 0 1 0	2
0 0 0 0 1 0 0	3
0 0 0 1 0 0 0	4
0 0 1 0 0 0 0	5
0 1 0 0 0 0 0	6
1 0 0 0 0 0 0	7

Q: How many variables are necessary to represent this function?

Distribution is Skewed

- For each x_i, there is single 1, and 6 0's.

1-out-of-7 code *x*7 *x*6*x*5 *x*4*x*3*x*2*x*1	Index
0 0 0 0 0 0 1	1
0 0 0 0 0 1 0	2
0 0 0 0 1 0 0	3
0 0 0 1 0 0 0	4
0 0 1 0 0 0 0	5
0 1 0 0 0 0 0	6
1 0 0 0 0 0 0	7

Only Three Variables are Necessary.

$$y_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7$$

$$y_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$$

$$y_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7$$

Original *vs.* Transformed Variables

$$y_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7$$

$$y_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$$

$$y_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7$$

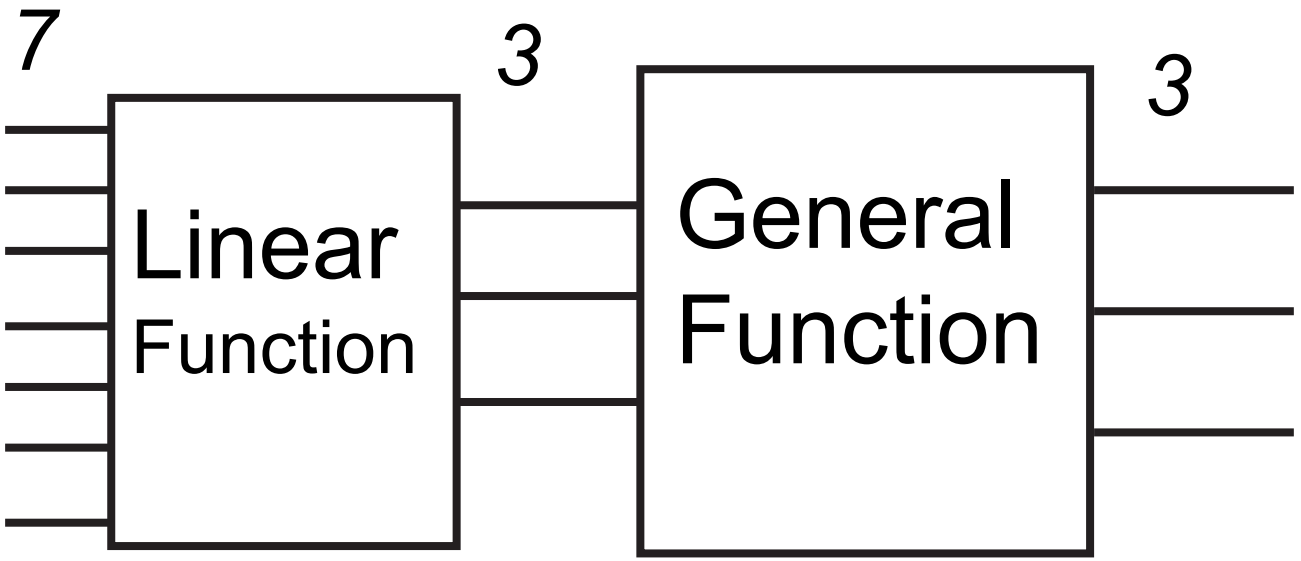
Transformed variables

For each of *y*1,*y2*,*y3*,

there are 3 0's, and 4 1's.

*x*7 *x*6*x*5 *x*4*x*3*x*2*x*1	y3 y2 y1
0 0 0 0 0 0 0 1	0 0 1
0 0 0 0 0 0 1 0	0 1 0
0 0 0 0 0 1 0 0	0 1 1
0 0 0 0 1 0 0 0	1 0 0
0 0 0 1 0 0 0 0	1 0 1
0 0 1 0 0 0 0 0	1 1 0
0 1 0 0 0 0 0 0	1 1 1
1 0 0 0 0 0 0 0	

Linear Decomposition



of Variables to Represent IP Address Tables (***n***=32)

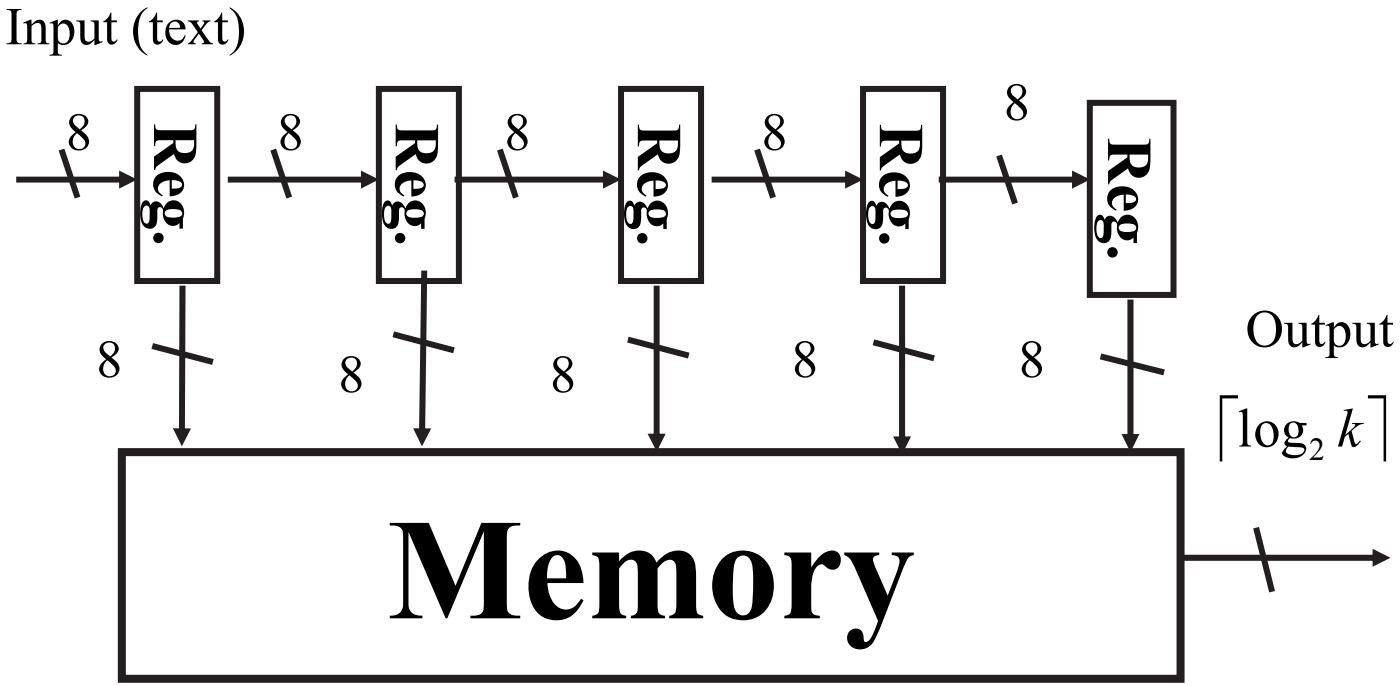
k	*t=1*	*t=2*	*t=3*
1670	**18**	**17**	**16**
3288	**20**	**19**	**18**
4591	**21**	**20**	**19**
7903	**23**	**21**	**20**

t **: Compound degree**

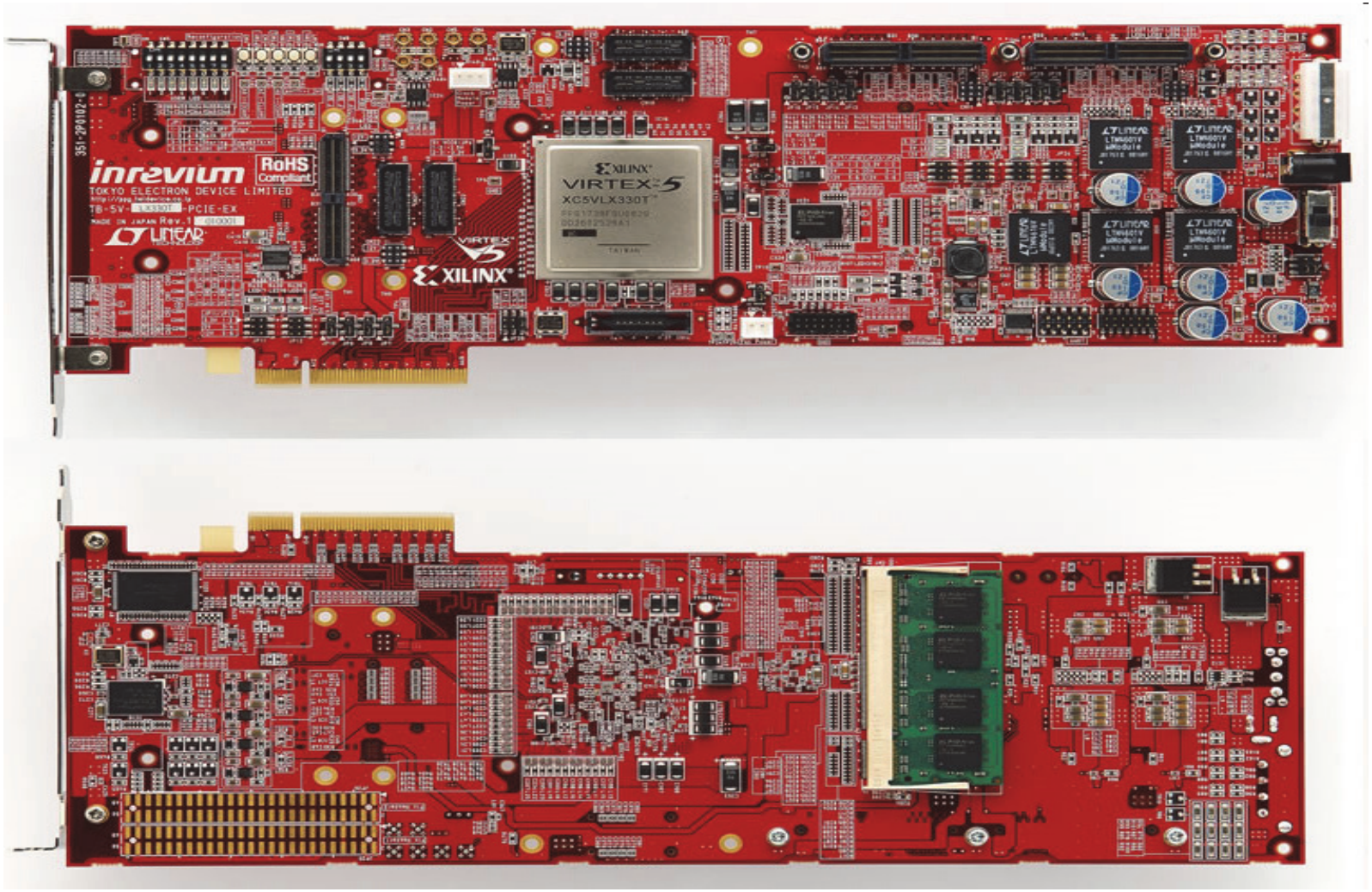
Scanning Engine for Computer Virus

- Stores k=1.3 million virus sub-patterns of n=40 bits.
- The single LUT realization requires 21 Tera bits.
- Our method requires only 160 Mega bits.

Circuit to Detect Suspicious Patterns



5 characters（40 bits）, 1.3 million patterns



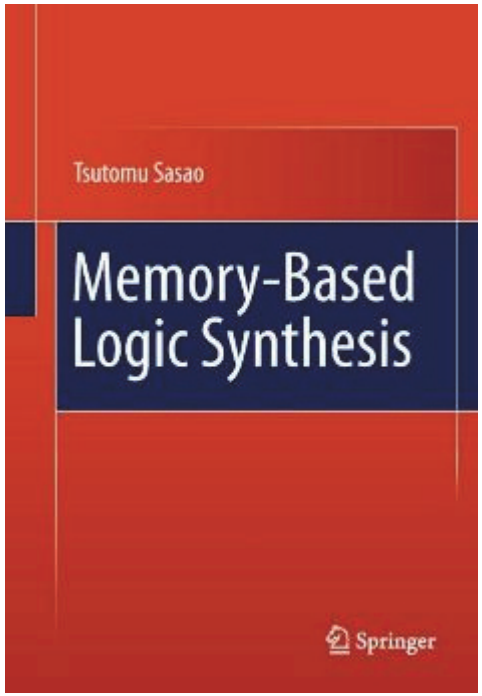
H. Nakahara, T. Sasao, M. Matsuura, "A low-cost and high-performance virus scanning engine using a binary CAM emulator and an MPU," 8th International Symposium on Applied Reconfigurable Computing, (ARC 2012), March 19-23, 2012, Hong-Kong.

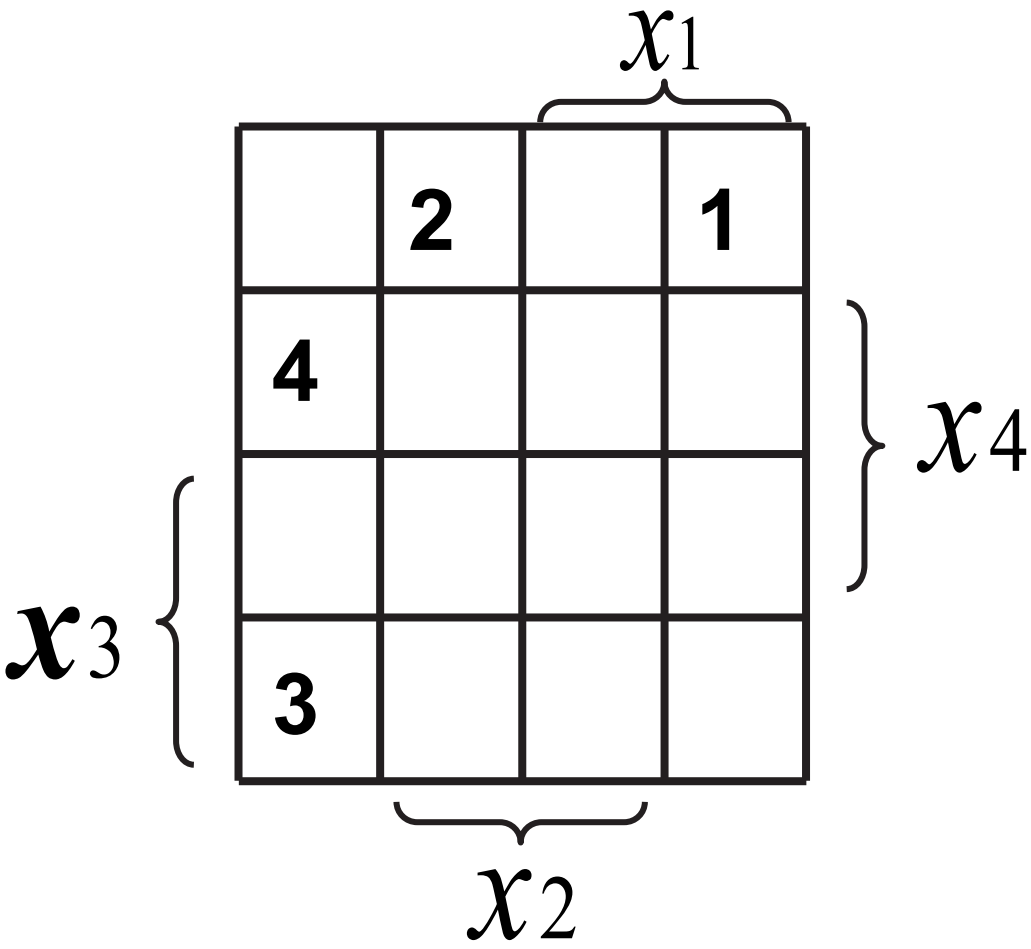
Current Projects

- Minimizers for input variables
  - high-speed applications
  - exact minimum
- Functional decomposition
  - using multiple IGUs
- New applications
  - Replacement for CAMs
  - Large-scale search engine

Reference

- T. Sasao, *Memory-Based Logic Synthesis*, Springer 2011.





x4 x3 x2 x1	Index
0 0 0 1	1
0 0 1 0	2
0 1 0 0	3
1 0 0 0	4