

CNS 188a Overview

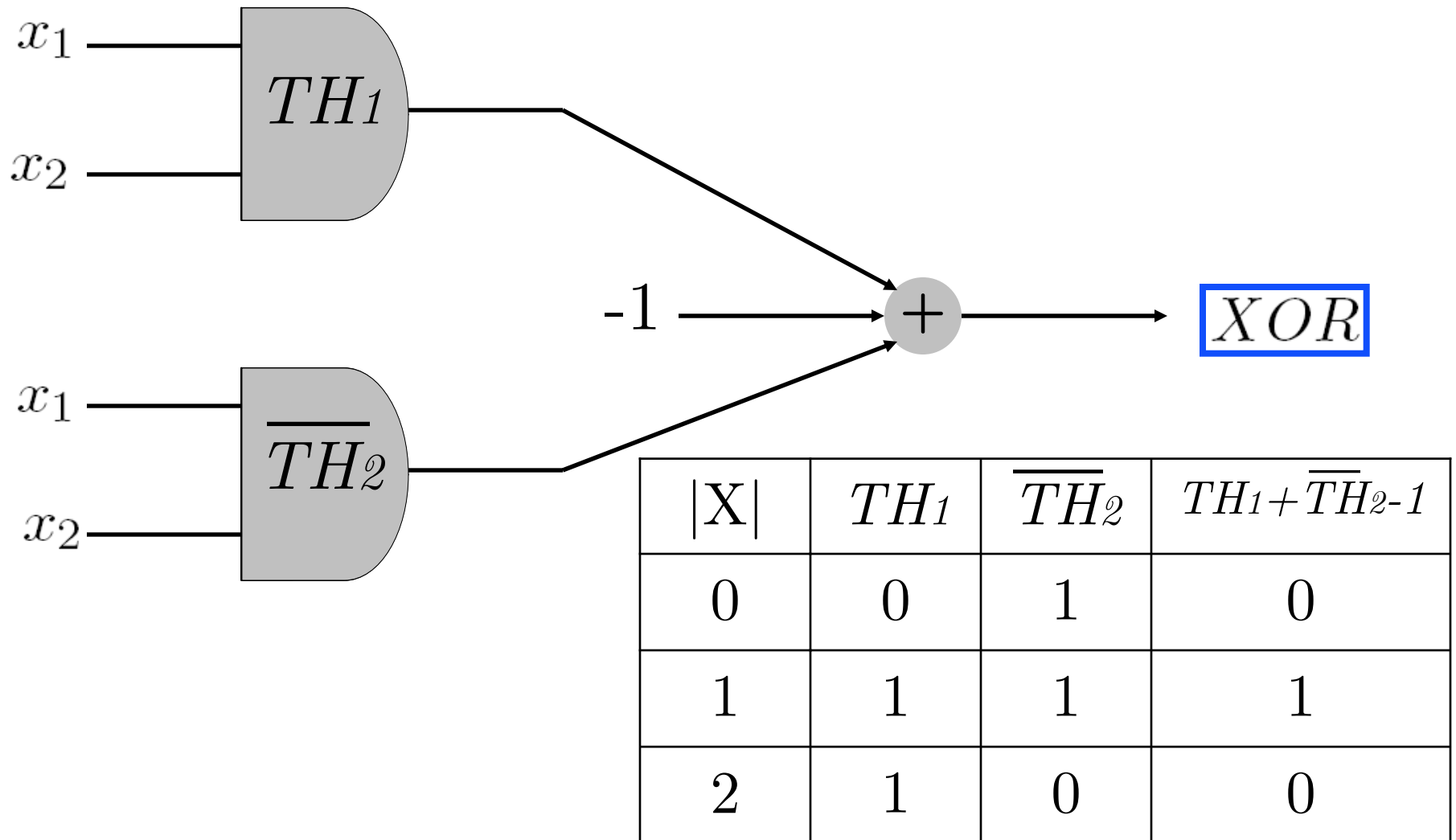
- Boolean algebra as an axiomatic system
- Boolean functions and their representations using Boolean formulas and spectral methods
- Implementing Boolean functions with relay circuits, circuits of AON (AND, OR, NOT) gates and LT (Linear Threshold) gates
- Analyzing the complexity (size and depth) of circuits
- Relations (as opposed to functions) and their implementation in circuits
- Feedback and convergence in LT circuits

XOR Function: Size of LT vs. AON in Depth 2

	$XOR(x_1, x_2, x_3)$	$XOR(x_1, x_2, \dots, x_n)$
AON	5	$2^{n-1} + 1$
LT-1	4	$n + 1$
LT-nl	2	$\left\lfloor \frac{n}{2} \right\rfloor + 1$

Symmetric functions

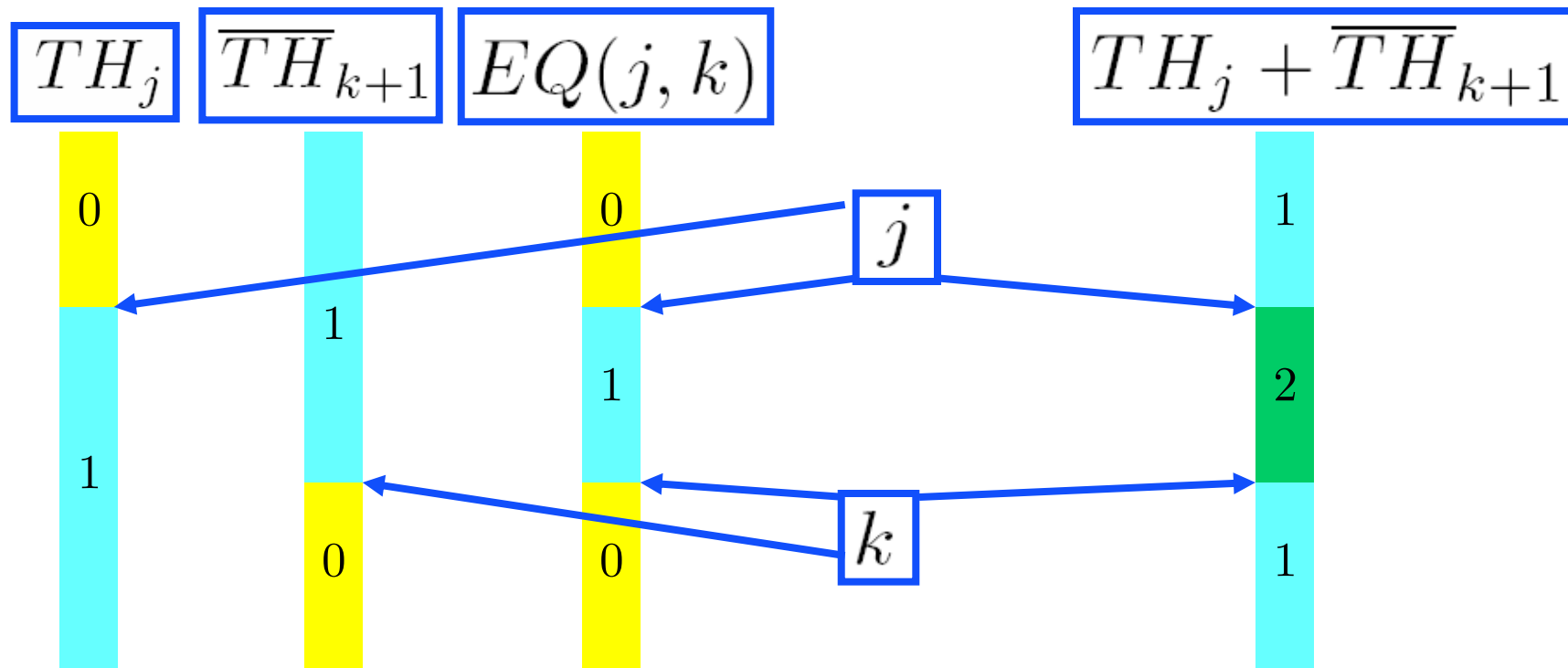
How is *SYM* related to LT_2 ?



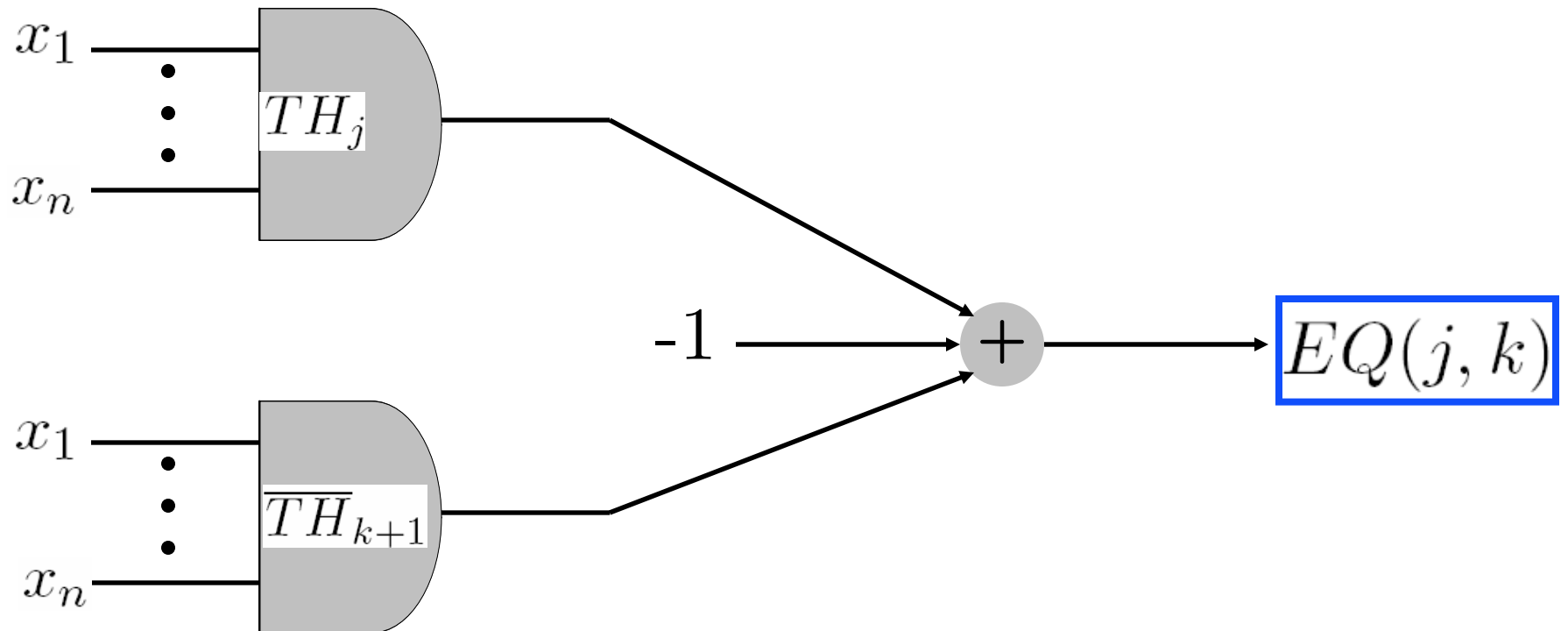
Generalization to EQ

$$EQ(j, k) = \begin{cases} 1 & \text{if } j \leq |X| \leq k \\ 0 & \text{else} \end{cases}$$

$$EQ(j, k) = TH_j + \overline{TH}_{k+1} - 1$$



Generalization to EQ



Q:

What is the generalization to arbitrary symmetric functions?

Generalization to SYM

Q:
What is the generalization to arbitrary **symmetric functions**?

A:
Consider the **symmetric function table**, it is a sum of non-overlapping 1-intervals

0

1

0

1

$$f(X) = \sum_{(j,k) \text{ 1-intervals}} EQ(j,k)$$

Sum of two TH functions

LT-I Circuit Design Algorithm for SYM

Muroga 1959

$ X $	$f(X)$	$f(X) = \overline{TH}_2 + TH_3 + \overline{TH}_5 + TH_6 - 1$	
0	1	$0 \Rightarrow 1$	$1 \Rightarrow 0$
1	1	$\overline{TH}_2$	TH_i
2	0		$\overline{TH}_i$
3	1	TH_3	
4	1	$\overline{TH}_5$	Subtract 1 for every isolated 1-block
5	0		
6	1	TH_6	
7	1		

LT-I Circuit Design Algorithm for SYM

Q: using the design algorithm,
how many TH gates are used the first layer?

0

1

0

1

Definition:

Let $T(f)$ be the number of transitions
in the symmetric function table

A: $T(f)$ TH gates are used
 $T(f) = 1$ is the single gate case

Can we do better?

The Layered Construction for SYM Is OPTIMAL

Summary of the Proof

Theorem (SYM with LT):

A layered LT circuit for a symmetric function f , with the first layer being TH gates, has at least $T(f)+1$ gates for $T(f)$ greater than 1.

Proof:

A circuit for a symmetric function (not in TH) has at least 3 gates

Given a layered LT circuit with k gates for a symmetric function f

There exists a symmetric function g , such that:

- $T(g) = T(f) - 1$
- The function g can be implemented by a layered LT circuit with $k-1$ gates

$T(f)$	k
2	3
3	3
4	4
5	5
6	6



Q

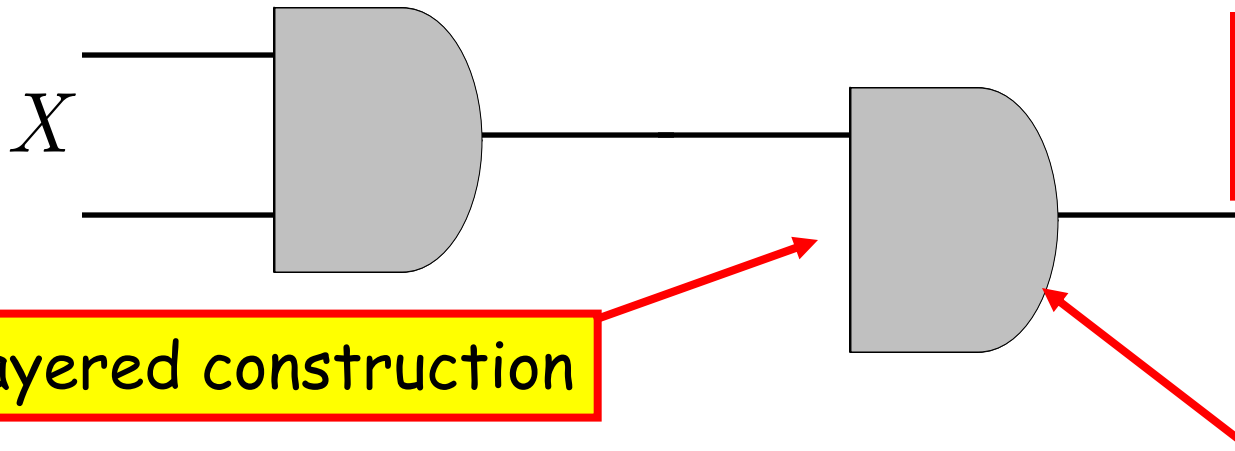
The Layered Construction for SYM Is OPTIMAL

Proof:

Idea: start with $T(f) = 2$

Need to prove: the LT circuit for f has at least 3 gates

f is not in TH, hence we need at least 2 LT gates



Need at least 3 gates!

Layered construction

Single input: identity or complement, hence this gate is redundant

The Layered Construction for SYM Is OPTIMAL

Say if we prove *: Given a layered LT circuit with k gates for a symmetric function f



There exists a symmetric function g , such that:

- $T(g) = T(f) - 1$
- The function g can be implemented by a layered LT circuit with $k-1$ gates

Then it follows that $k \geq T(f) + 1$

Why?

$T(f)$	k (circuit size)
2	3
3	3
4	4
5	5
6	6

By *



Contradiction

The Layered Construction for SYM Is OPTIMAL

Need to prove:

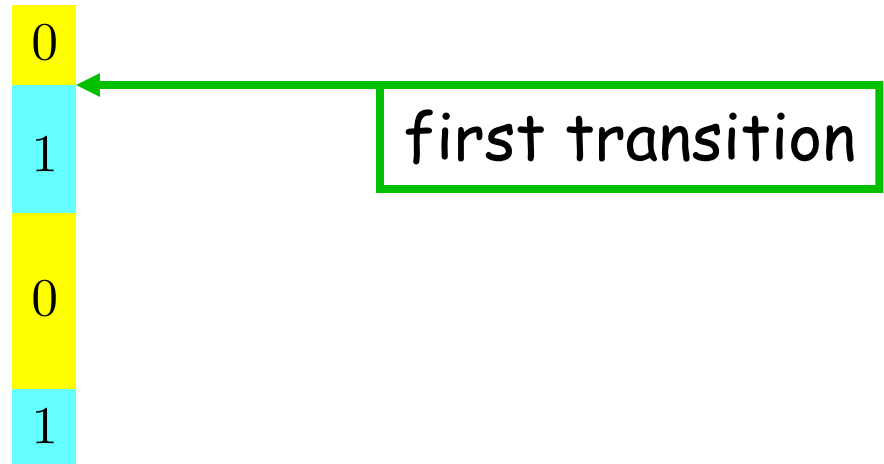
Given a layered LT circuit with k gates for a symmetric function f



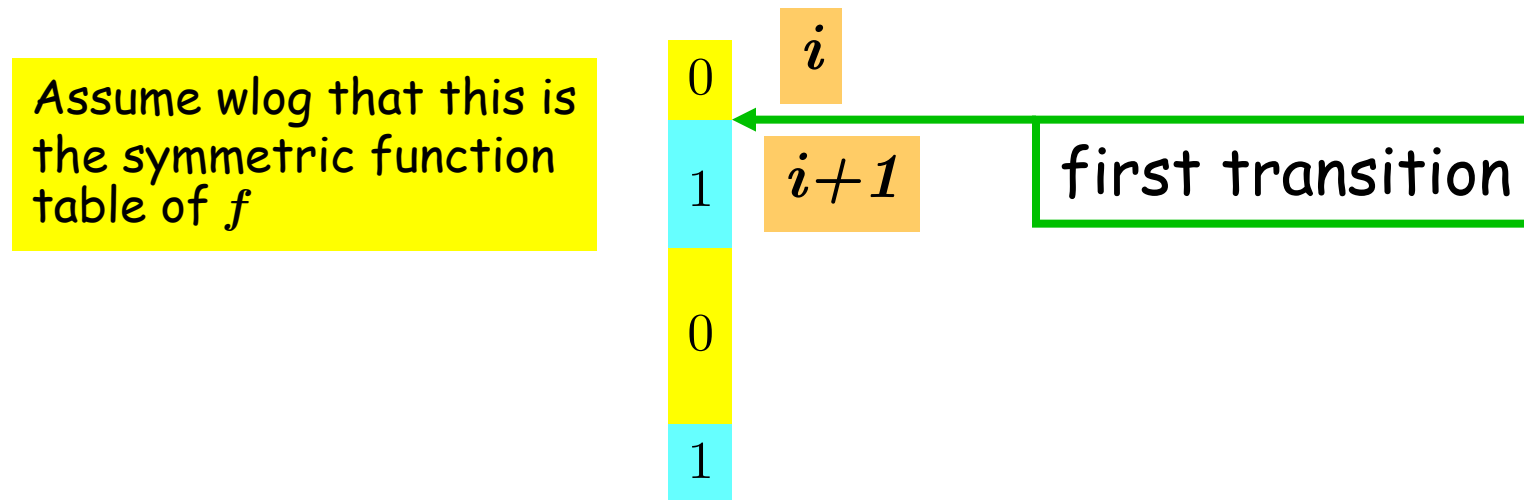
There exists a symmetric function g , such that:

- $T(g) = T(f) - 1$
- The function g can be implemented by a layered LT circuit with $k-1$ gates

Assume wlog that this is the symmetric function table of f



The Layered Construction for SYM Is OPTIMAL



Claim: There must be an **TH gate** in the first layer that detects the change from $|X| = i$ to $|X| = i+1$

Proof:

Note that it is a layered circuit, namely, inputs are connected only to the first layer!!

The Layered Construction for SYM Is OPTIMAL

Claim: There must be an **TH gate** in the first layer that detects the change from $|X| = i$ to $|X| = i+1$

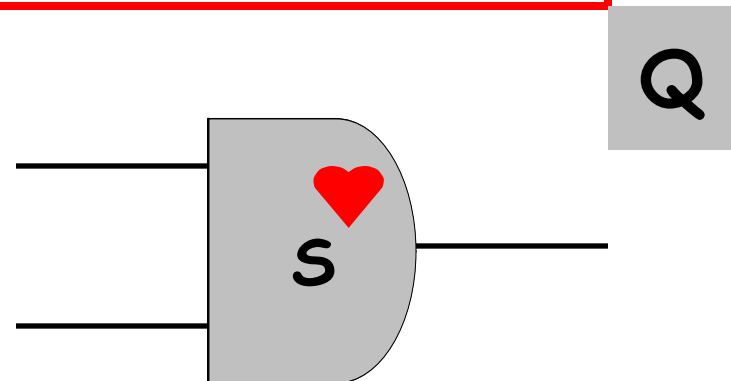
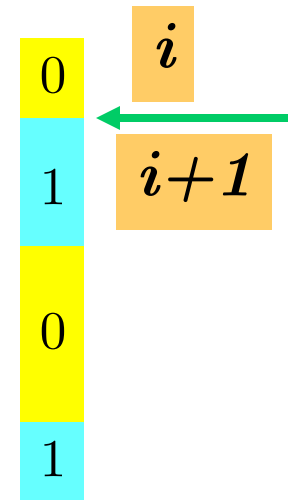
Proof:

It is a layered circuit, namely, inputs are connected only to the first layer!!

If all TH gates in the first layer output the same for $|X| = i$ and $|X| = i+1$ then the circuit **will not compute f** :
$$f(|X| = i) = f(|X| = i + 1)$$

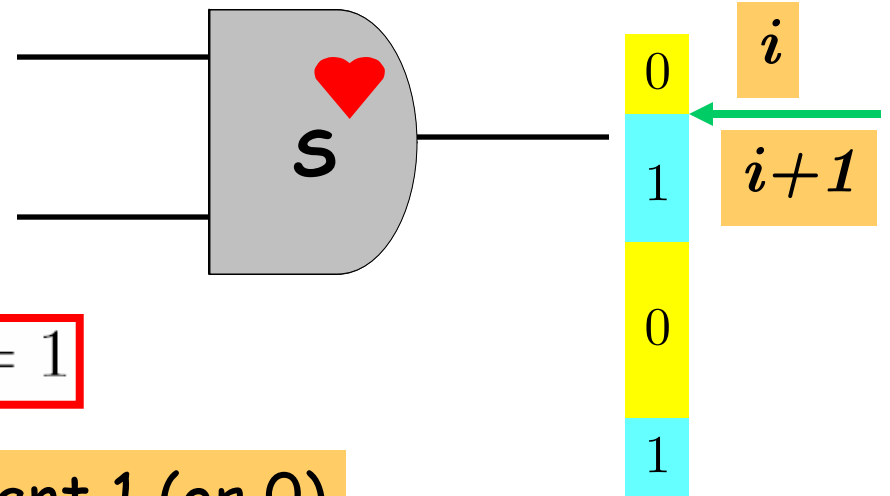
Call this "sensitive" gate **S**

Can we eliminate **S** ??



The Layered Construction for SYM Is OPTIMAL

Can we eliminate S ??



Easy: $x_1 = x_2 = \dots = x_{i+1} = 1$

The output of S will be a constant 1 (or 0)

Setting the variables to $x_1 = x_2 = \dots = x_{i+1} = 1$ results in a **new circuit** and a **new function**

Q: are we done?

The Layered Construction for SYM Is OPTIMAL

Given a layered LT circuit with k gates for a symmetric function f



There exists a symmetric function g , such that:

- $T(g) = T(f) - 1$
- The function g can be implemented by a layered LT circuit with $k-1$ gates

Setting the variables to $x_1 = x_2 = \dots = x_{i+1} = 1$ results in a **new circuit** and a **new function**

- g is a symmetric function

- $T(g) = T(f) - 1$

g

- We removed the gate S (or more), hence, the circuit for g is layered and has **at most** $k-1$ gates

The Layered Construction for SYM Is OPTIMAL

Remarks

Theorem (SYM with LT):

A **layered** LT circuit for a symmetric function f , with the first layer being TH gates, has at least $T(f)+1$ gates for $T(f)$ greater than 1.

R1: circuit's depth

- Our proof works for arbitrary depth
- Depth-2 has optimal size
- Depth does not help in reducing the size (**layered**, TH)

R2: Open problems

- A lower bound for arbitrary LT gates (beyond TH)
- Or, a better layered construction using LT gates

XOR Function: Size of LT vs. AON in Depth 2

	$XOR(x_1, x_2, x_3)$	$XOR(x_1, x_2, \dots, x_n)$
AON	5	$2^{n-1} + 1$
LT-1	4	$n + 1$
LT-nl	2	$\left\lfloor \frac{n}{2} \right\rfloor + 1$

Symmetric functions; $T(f)$ TH gates



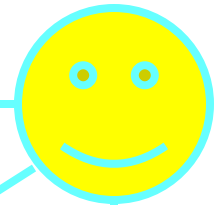
optimal

XOR Function: Size of LT vs. AON in Depth 2

	$XOR(x_1, x_2, x_3)$	$XOR(x_1, x_2, \dots, x_n)$
AON	5	$2^{n-1} + 1$
LT-I	4	$n + 1$
LT-nl	2	$\left\lfloor \frac{n}{2} \right\rfloor + 1$

Symmetric functions

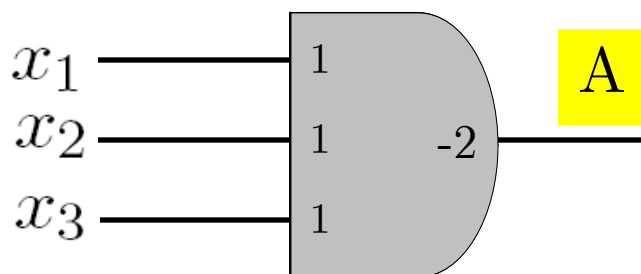
optimal



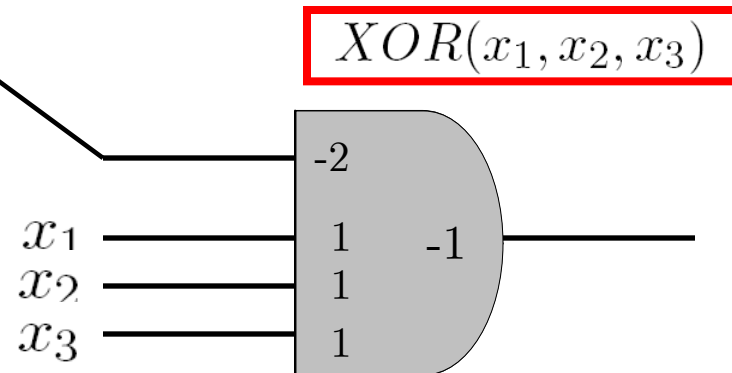
LT-nl Construction

$$T(f)+1 = 4$$

LT-nl = LT non-layered,
inputs go to any layer



$ X $	A	$-1-2A+ X $	$sgn(\quad)$
0	0	-1	0
1	0	0	1
2	1	-1	0
3	1	0	1



LT-nl Construction

Given a symmetric function f with k 1s in the symmetric function table

At locations: $q_1, q_2, q_3, \dots, q_k$

Construct an LT-nl circuit with $k+1$ gates

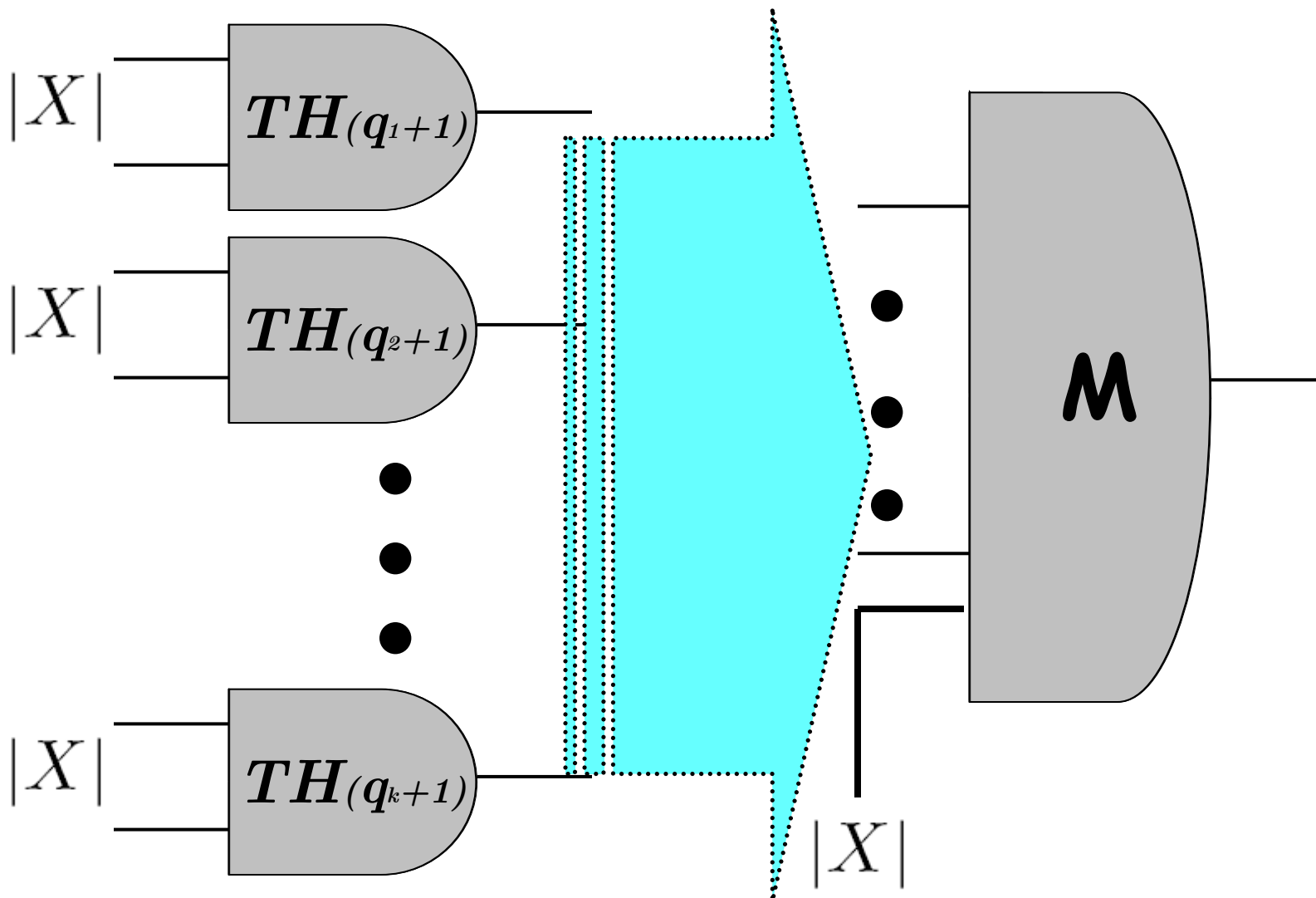
$$k = 5$$

$$\begin{aligned} q_1 &= 1 \\ q_2 &= 3 \\ q_3 &= 4 \\ q_4 &= 6 \\ q_5 &= 7 \end{aligned}$$

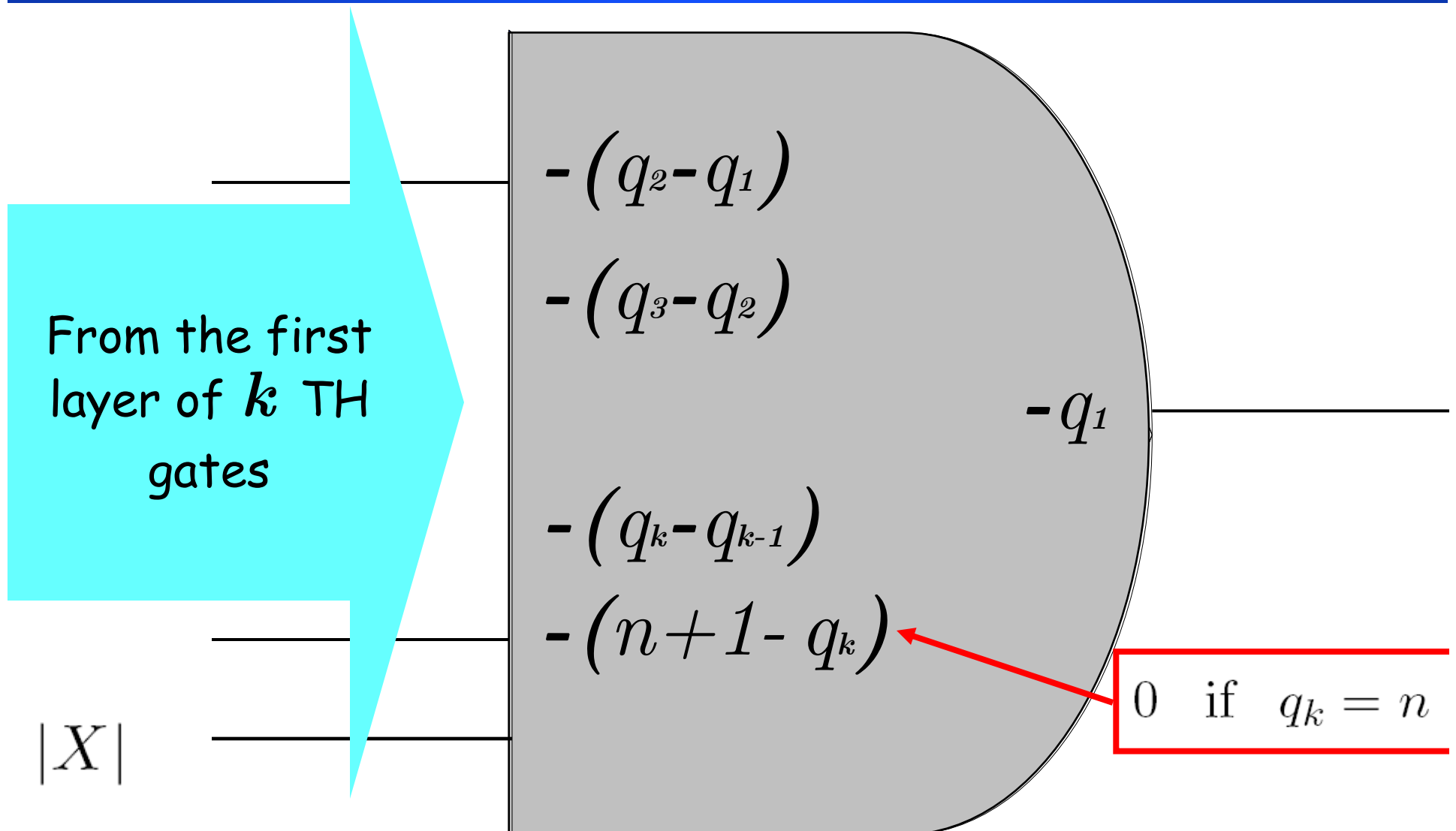
$ X $	$f(X)$
0	0
1	1
2	0
3	1
4	1
5	0
6	1
7	1

LT-nl Construction

Minnick 1961



The Gate M



LT-nl Construction - Example

Given a symmetric function f with k 1s in the symmetric function table

At locations: $q_1, q_2, q_3, \dots, q_k$

Construct an LT-nl circuit with $k+1$ gates

$$k = 3$$

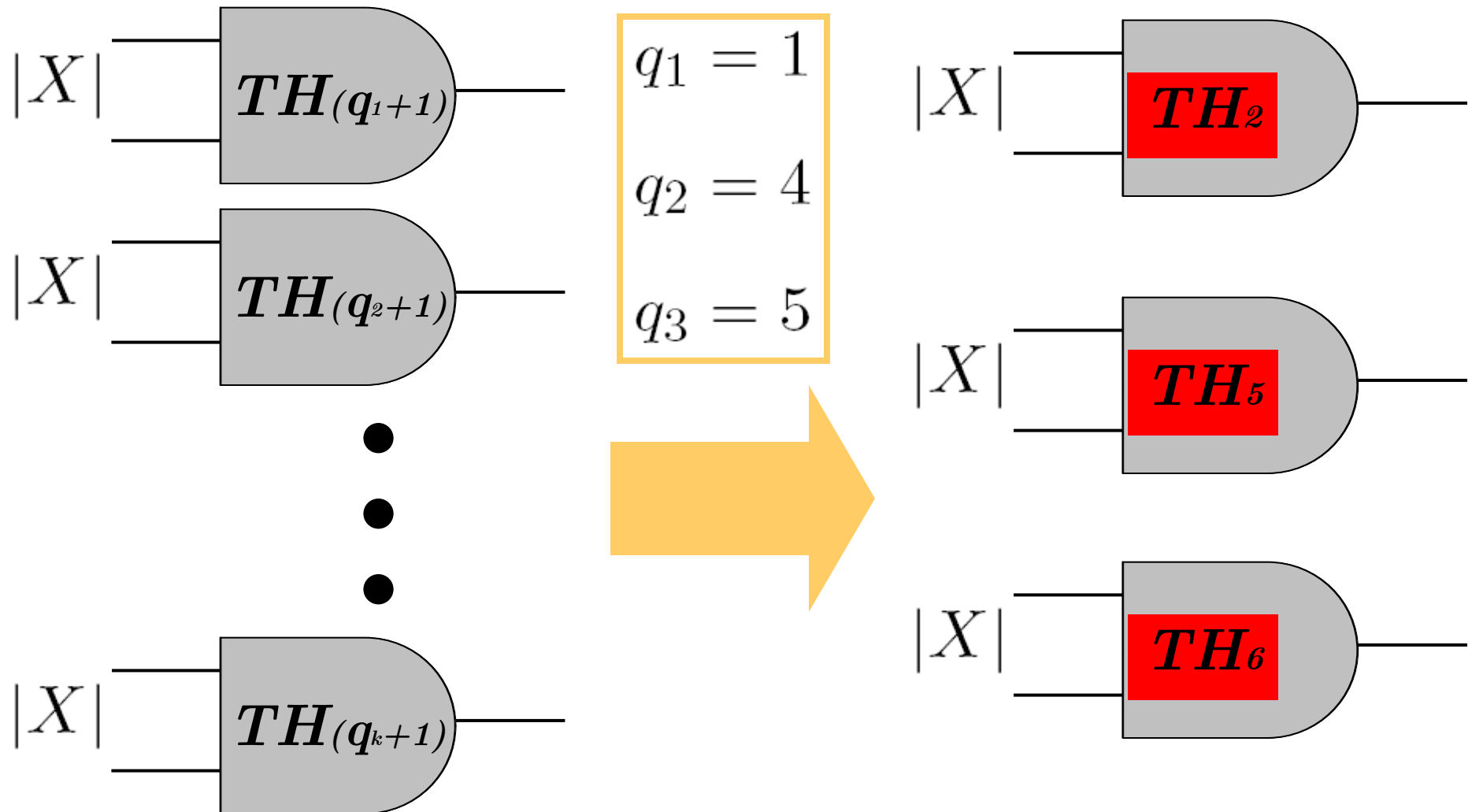
$$q_1 = 1$$

$$q_2 = 4$$

$$q_3 = 5$$

$ X $	$f(X)$
0	0
1	1
2	0
3	0
4	1
5	1

LT-nl Construction - First Layer



$$q_1 = 1$$

$$q_2 = 4$$

$$q_3 = 5$$

The Gate M

$$-(q_2 - q_1) = -3$$

$$-(q_3 - q_2) = -1$$

$$-q_1 = -1$$

$$0$$

$$0 \quad \text{if} \quad q_k = n$$

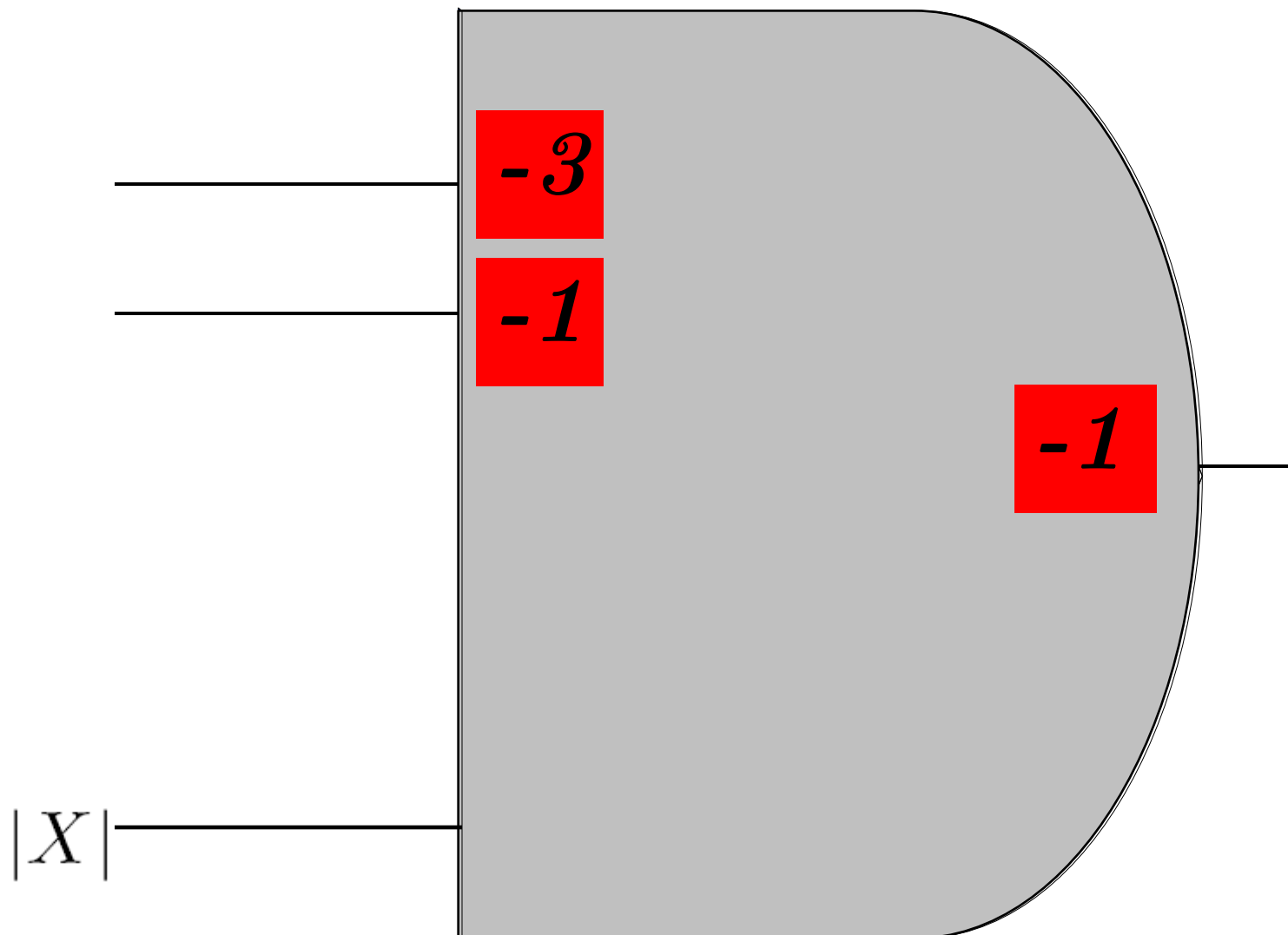
$|X|$

$$q_1 = 1$$

$$q_2 = 4$$

$$q_3 = 5$$

The Gate M

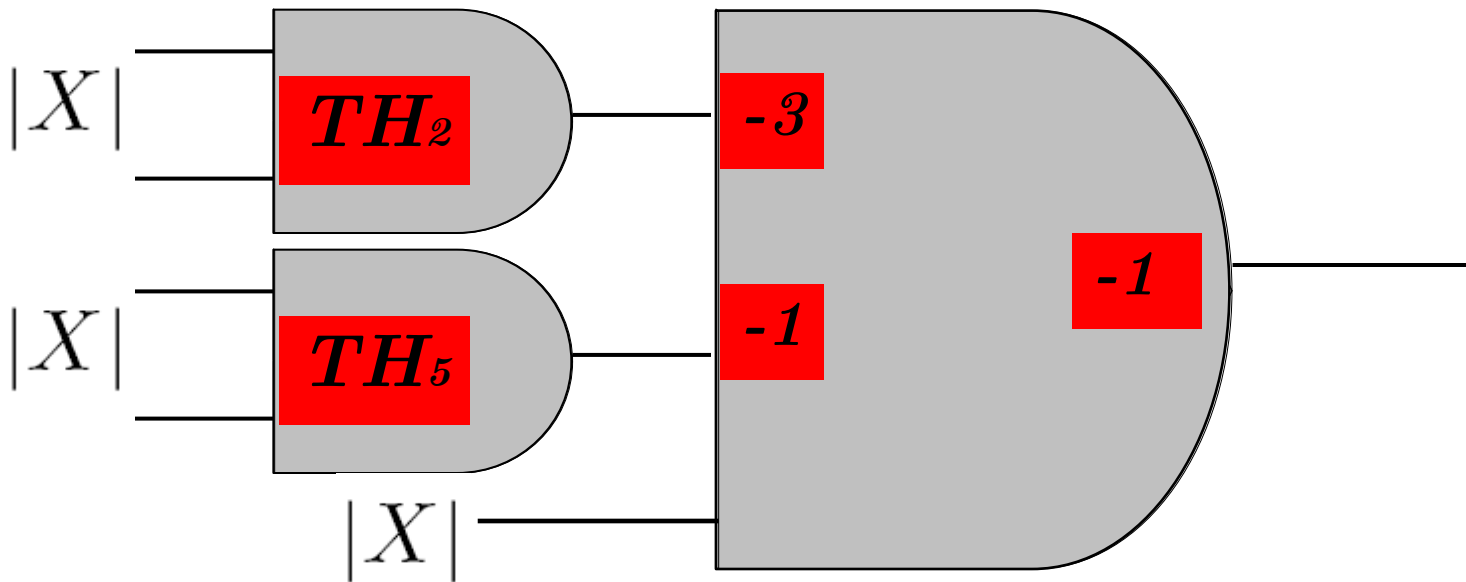


$$q_1 = 1$$

$$q_2 = 4$$

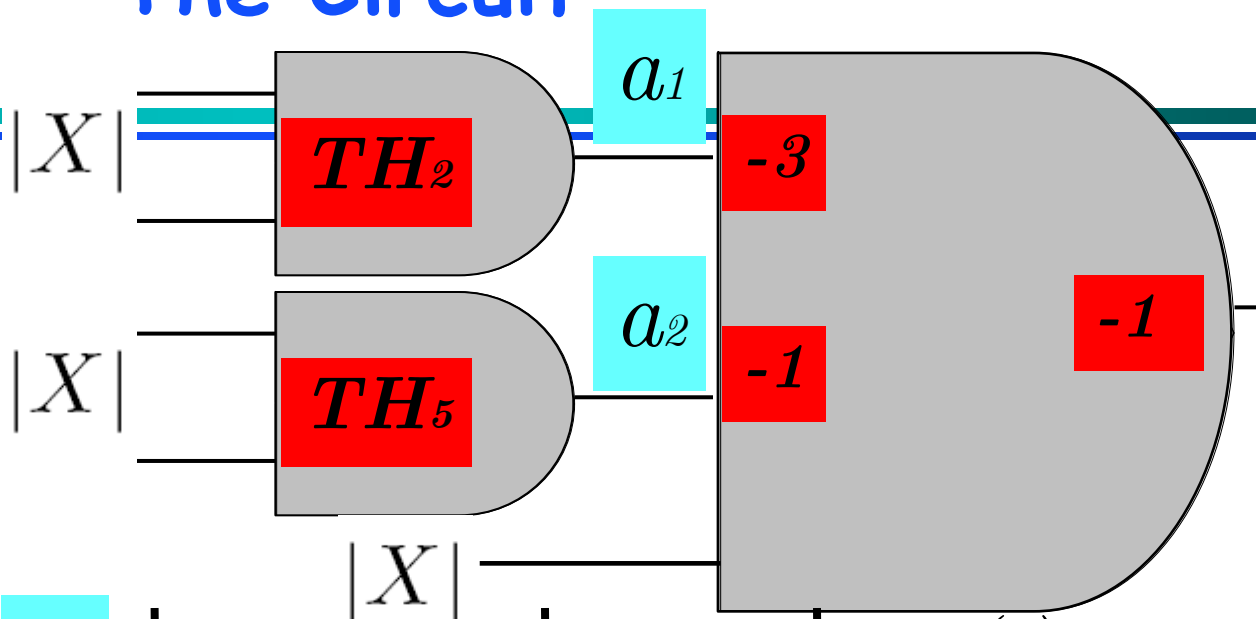
$$q_3 = 5$$

The Circuit



$$\begin{aligned} q_1 &= 1 \\ q_2 &= 4 \\ q_3 &= 5 \end{aligned}$$

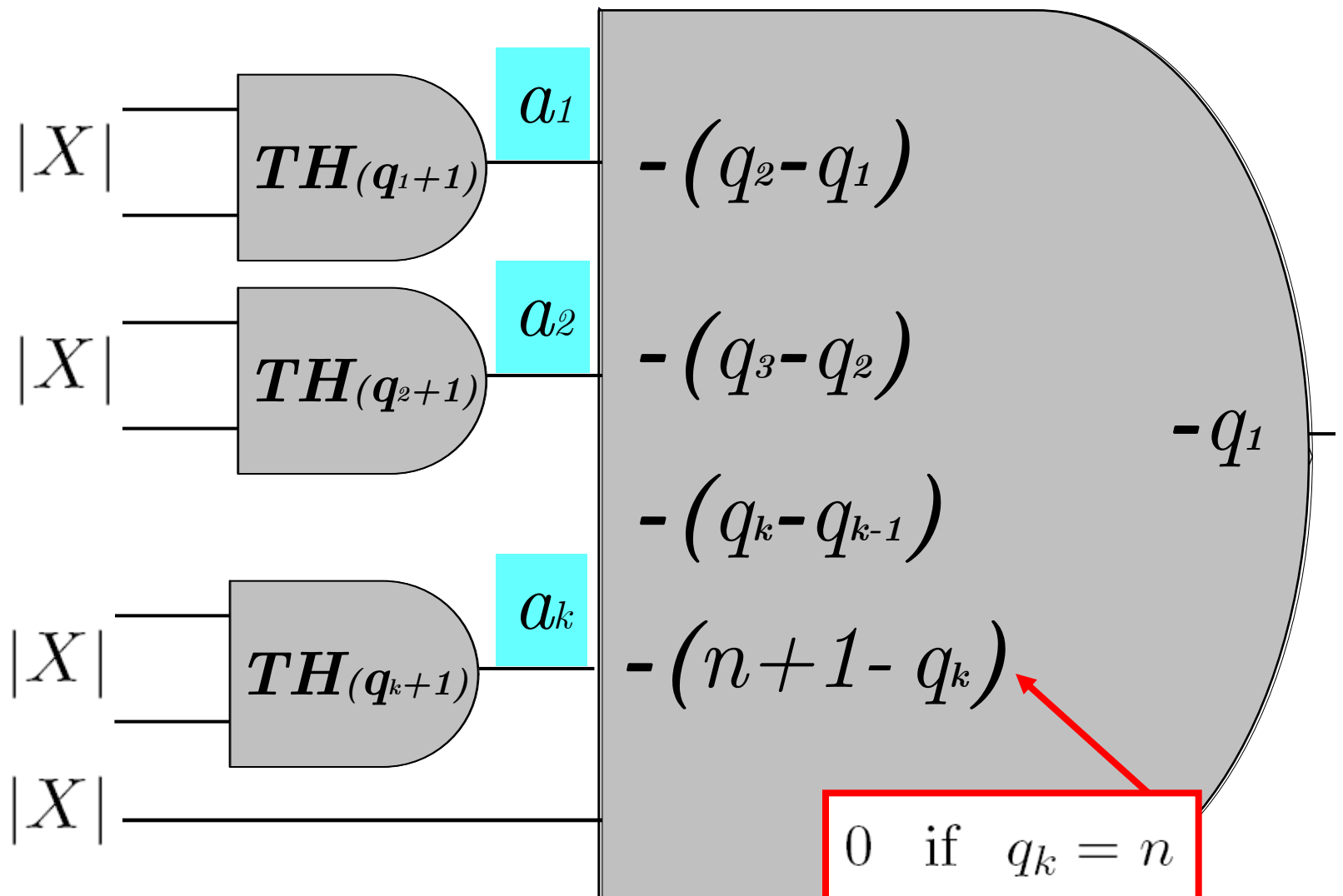
The Circuit



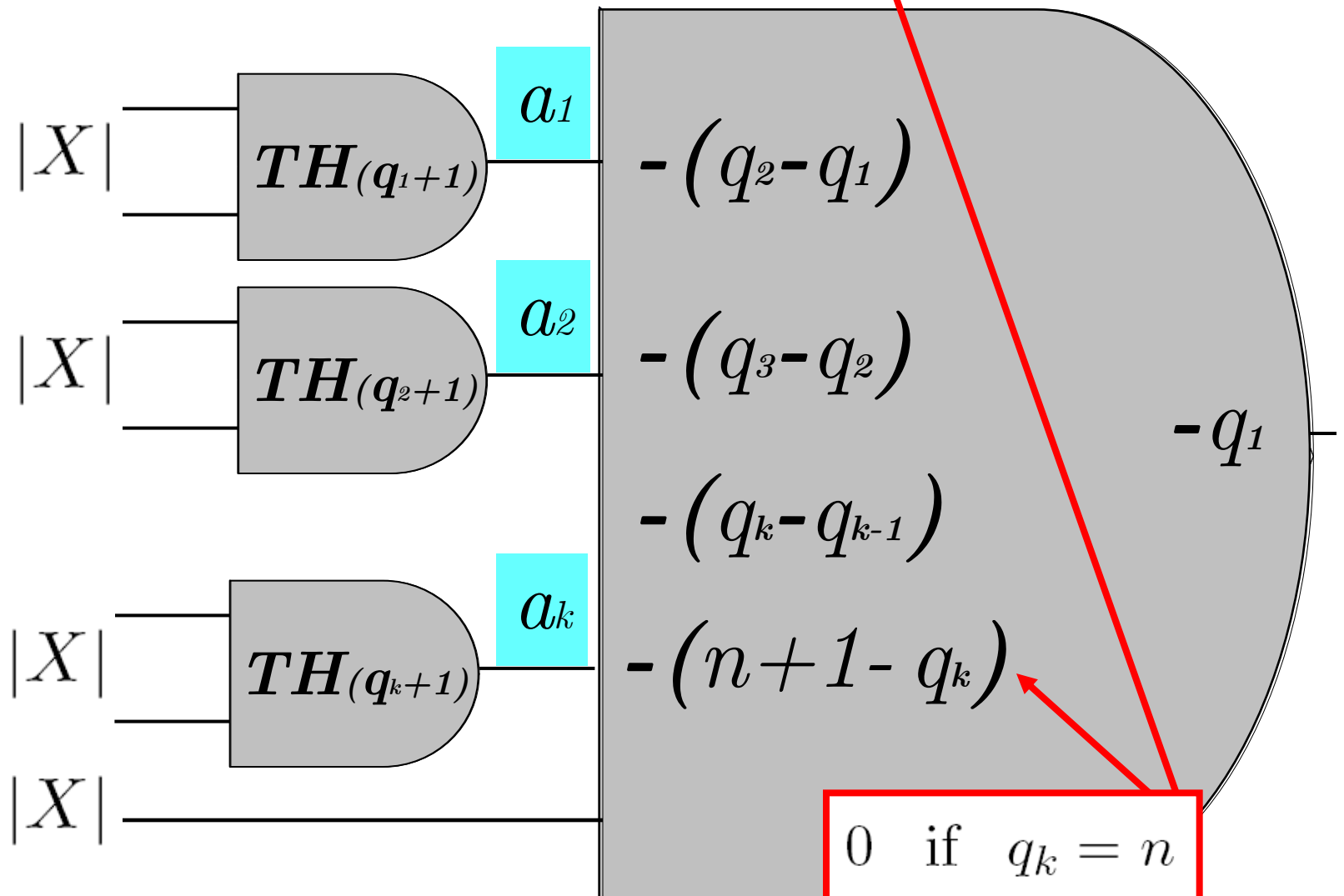
$ X $	$f(X)$	a_1	a_2	$-3a_1 - a_2 - 1$	$\dots + X $	$sgn(\cdot)$
0	0	0	0	-1	-1	0
1	1	0	0	-1	0	1
2	0	1	0	-4	-2	0
3	0	1	0	-4	-1	0
4	1	1	0	-4	0	1
5	1	1	1	-5	0	1

The Minnick Construction

Why Does it work?

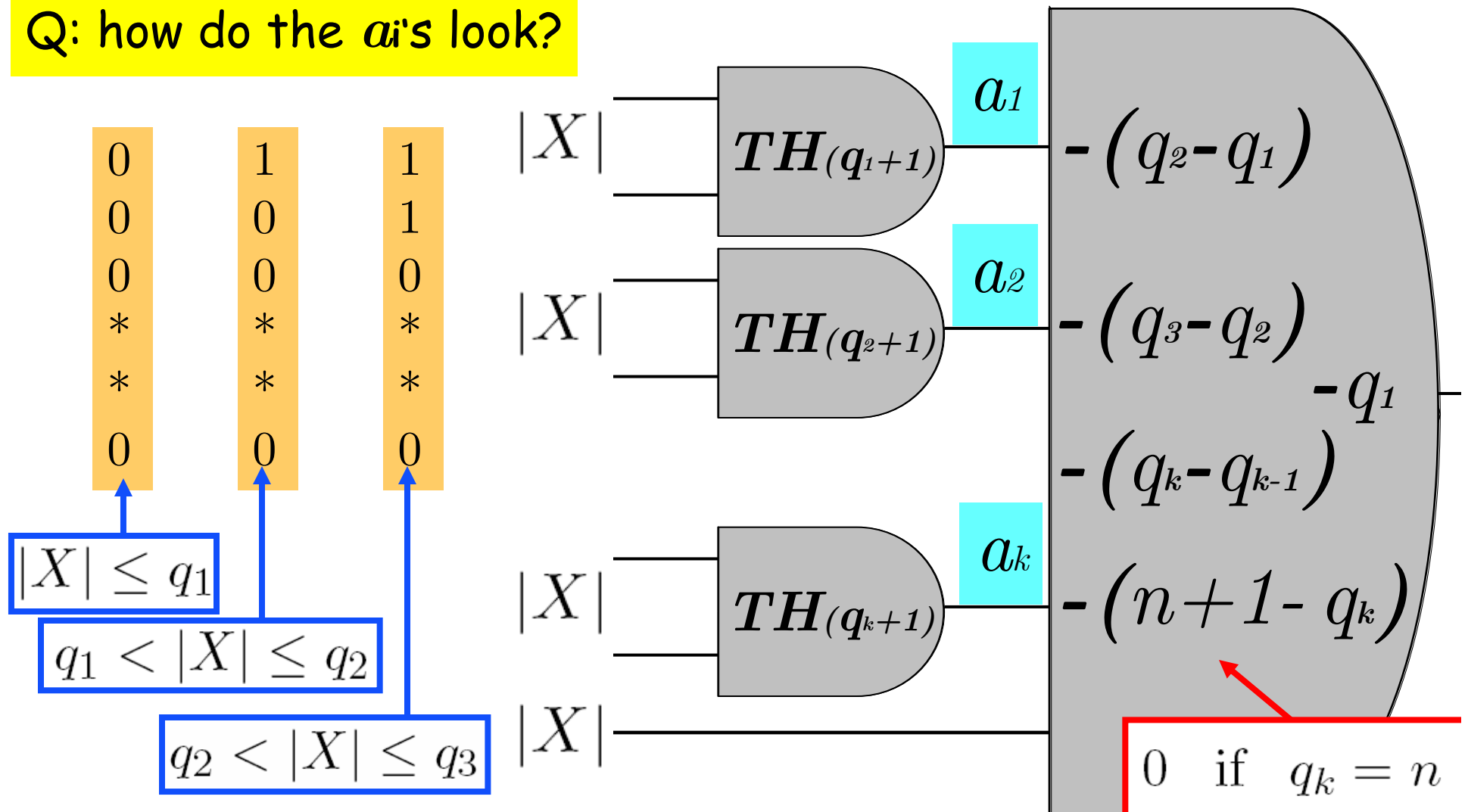


$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$

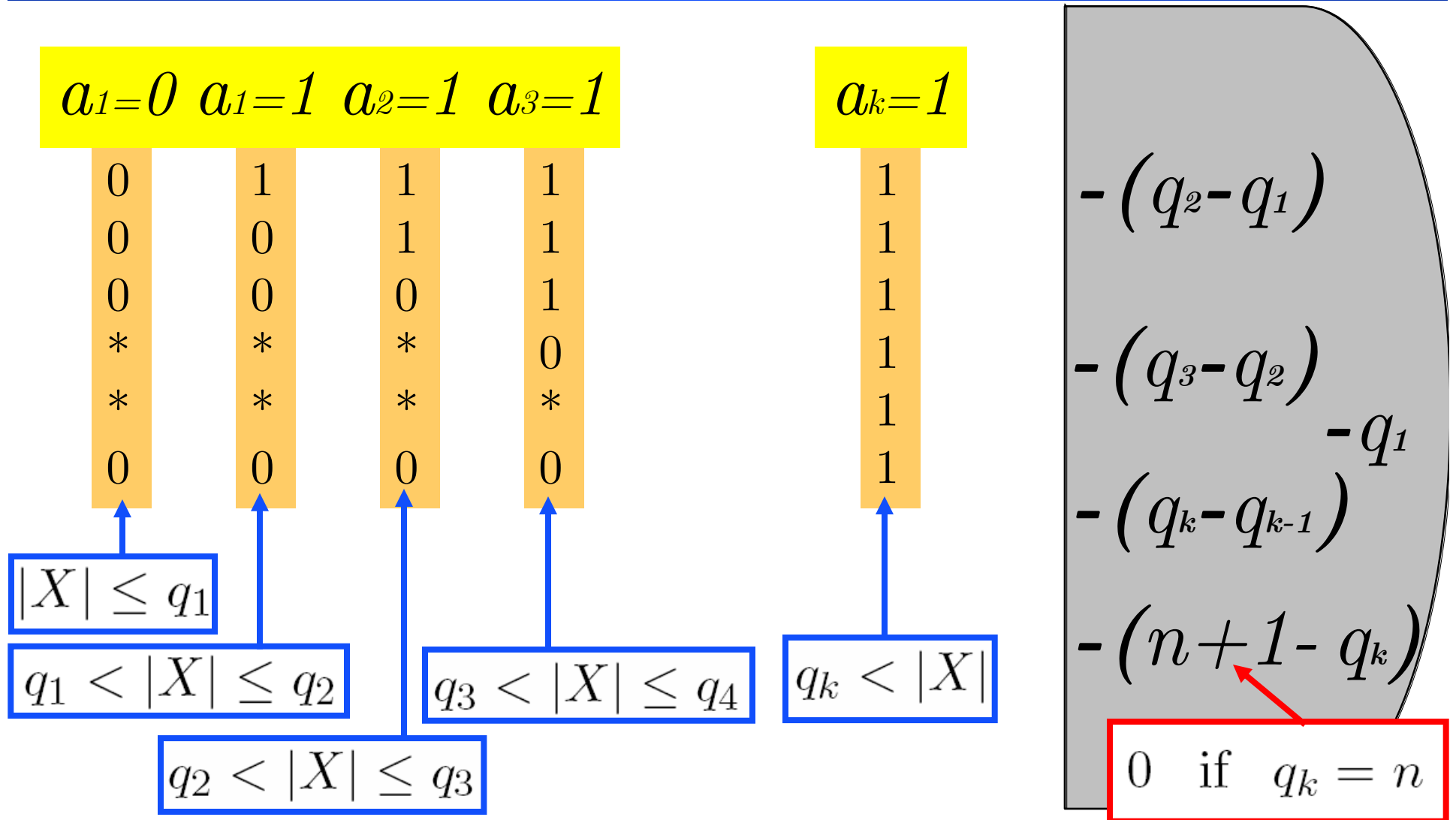


$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$

Q: how do the a_i 's look?



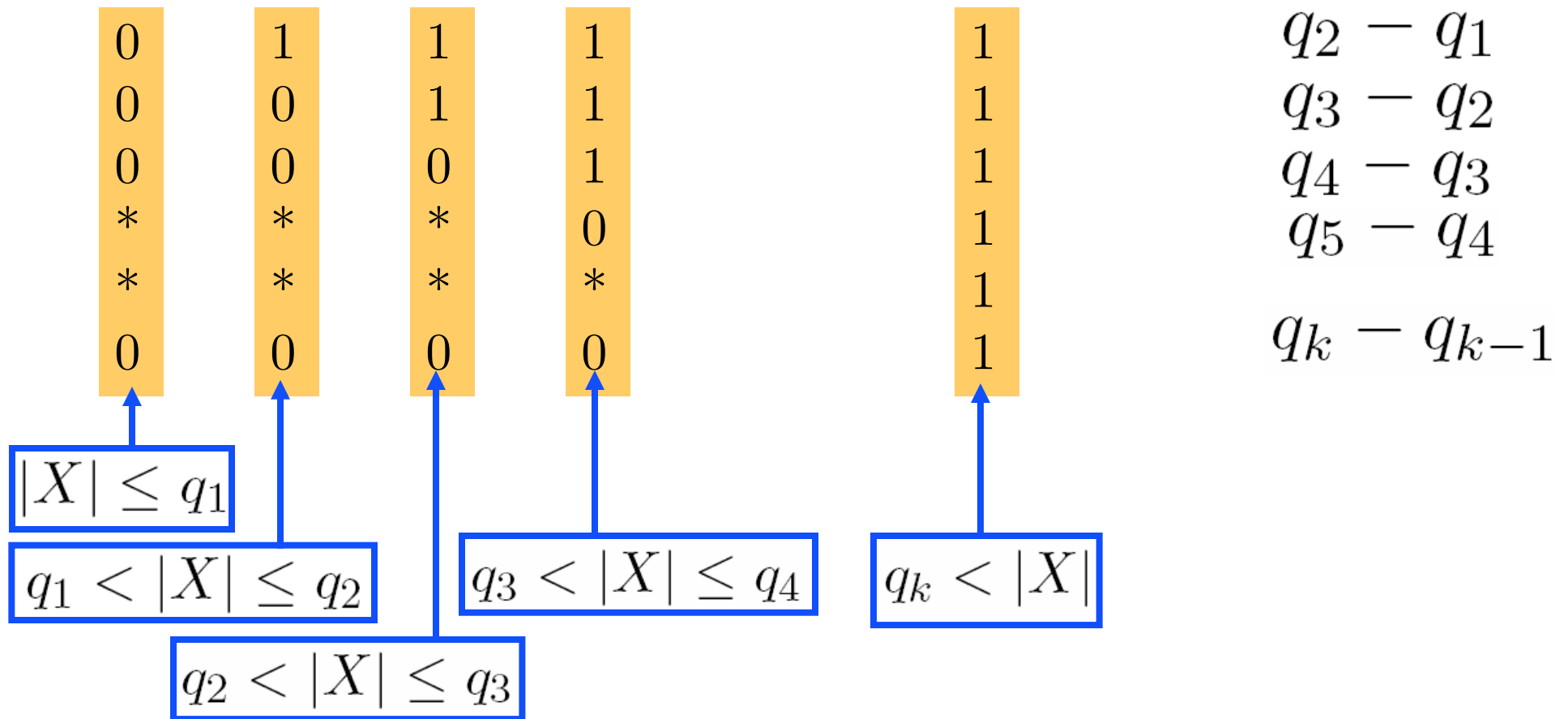
$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$



$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$

???

What was Minnick's key idea?



The Key: Telescopic Sum

$$q_4 < |X| \leq q_5$$

$$a_1 = 1$$

$$a_2 = 1$$

$$a_3 = 1$$

$$a_4 = 1$$

$$a_5 = 0$$

1

1

1

1

0

0

*

0

~~q_1~~

~~$q_2 - q_1$~~

~~$q_3 - q_2$~~

~~$q_4 - q_3$~~

~~$q_5 - q_4$~~

$$F(X) = -q_5 + |X|$$

We are always left with the last element in the sum

$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$

The Key: Telescopic Sum

$$q_4 < |X| \leq q_5$$

$$\begin{array}{l} a_1=1 \\ a_2=1 \\ a_3=1 \\ a_4=1 \\ a_5=0 \end{array} \begin{array}{l} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ * \\ 0 \end{array}$$

$$\begin{array}{l} q_1 \\ q_2 - q_1 \\ q_3 - q_2 \\ q_4 - q_3 \\ q_5 - q_4 \end{array}$$

$$F(X) = -q_5 + |X|$$

$$\text{if } |X| = q_5 \text{ then } F(X) = 0 \implies f(x) = 1$$

$$\text{if } q_4 < |X| < q_5 \text{ then } F(X) < 0 \implies f(x) = 0$$

$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$

The Minnick Construction

Claim: The Minnick construction is correct

Proof:

$$F(X) = - \left(q_1 + \sum_{i=1}^{k-1} a_i (q_{i+1} - q_i) + a_k (n + 1 - q_k) \right) + |X|$$

Consider $|X|$ in each possible range:

$F(X)$ is 0 for all the upper bounds of the ranges and negative otherwise

Q

$$\begin{array}{ll} 0 \leq |X| \leq q_1 & F(X) = -q_1 + |X| \\ q_1 < |X| \leq q_2 & F(X) = -q_2 + |X| \\ q_2 < |X| \leq q_3 & F(X) = -q_3 + |X| \\ q_3 < |X| \leq q_4 & F(X) = -q_4 + |X| \\ \vdots & \\ \vdots & \\ q_{k-1} < |X| \leq q_k & F(X) = -q_k + |X| \end{array}$$

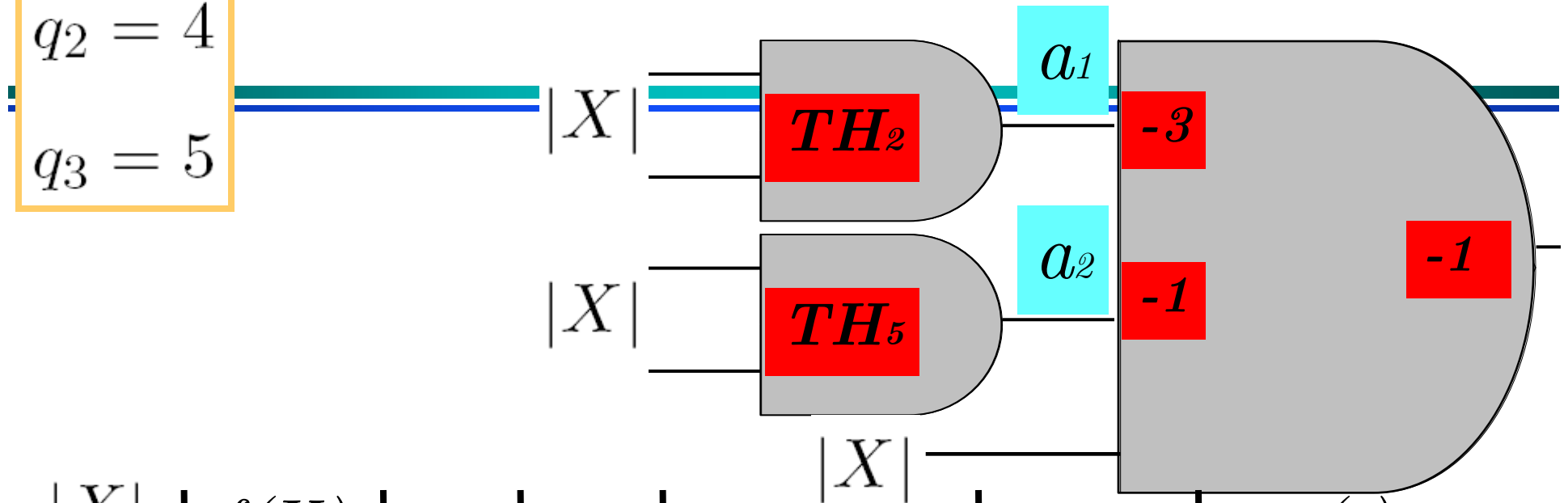
Telescopic sum

$$q_1 = 1$$

$$q_2 = 4$$

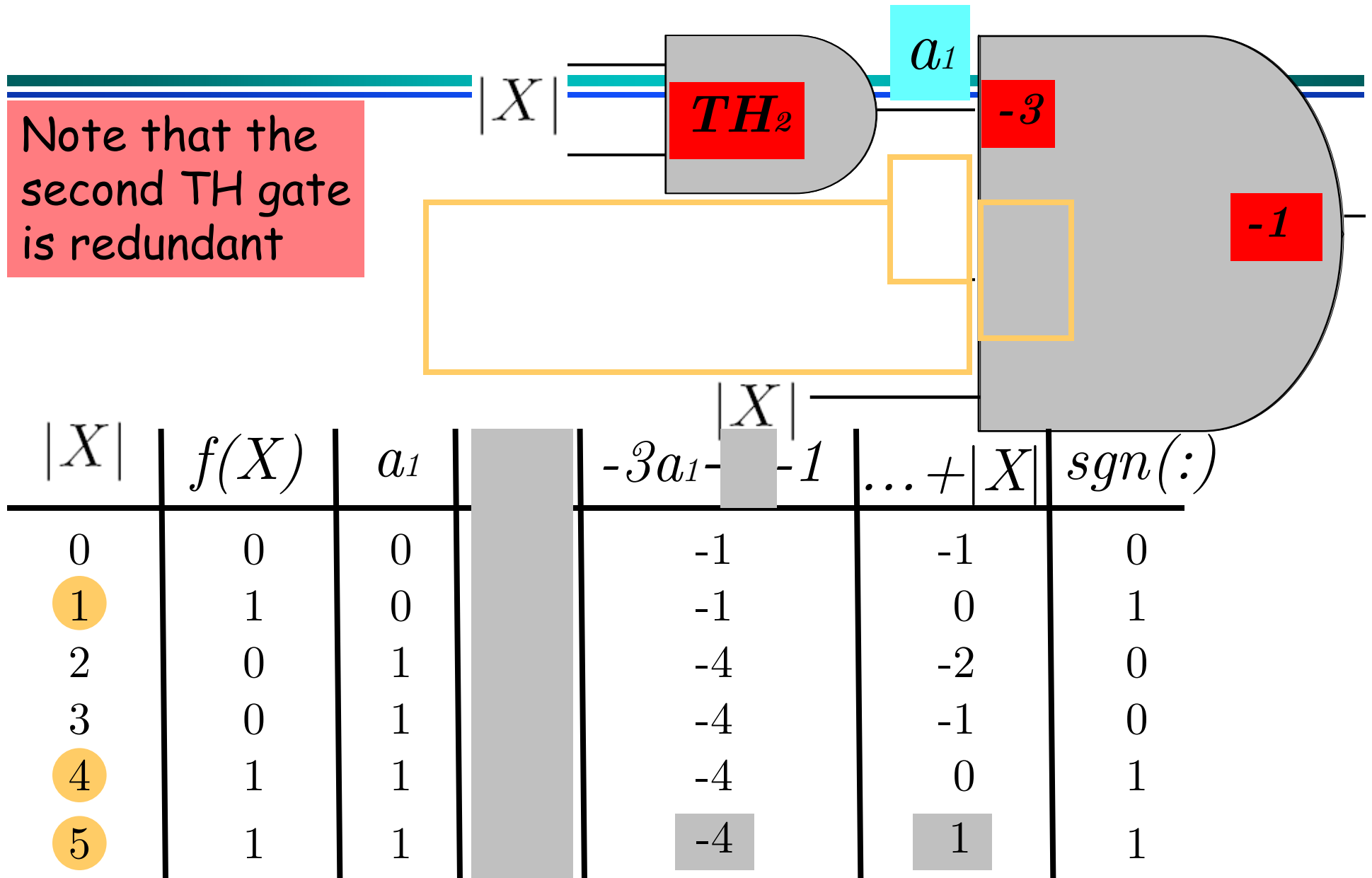
$$q_3 = 5$$

The Example Circuit



$ X $	$f(X)$	a_1	a_2	$-3a_1 - a_2 - 1$	$\dots + X $	$sgn(\cdot)$
0	0	0	0	-1	-1	0
1	1	0	0	-1	0	1
2	0	1	0	-4	-2	0
3	0	1	0	-4	-1	0
4	1	1	0	-4	0	1
5	1	1	1	-5	0	1

The Example Circuit



The Minnick Construction

Remarks

R1: The Minnick construction is **more general** than what we described, it works on **intervals**; we recommend that you read his paper (it is on the class web site)

R2: The Minnick construction is **optimal** for LT-nl of depth 2
We will not prove it in the lectures

XOR Function: Size of LT vs. AON in Depth 2

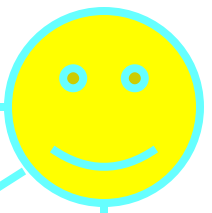
	$XOR(x_1, x_2, x_3)$	$XOR(x_1, x_2, \dots, x_n)$
AON	5	$2^{n-1} + 1$
LT-I	4	$n + 1$
LT-nl	2	$\left\lfloor \frac{n}{2} \right\rfloor + 1$

Symmetric functions

transitions

intervals

optimal



Questions on LT Constructions

Q1: we understand how to implement a **single** symmetric function, what if we need to implement a circuit that computes a **number** of symmetric functions?
Can we save in circuit size?

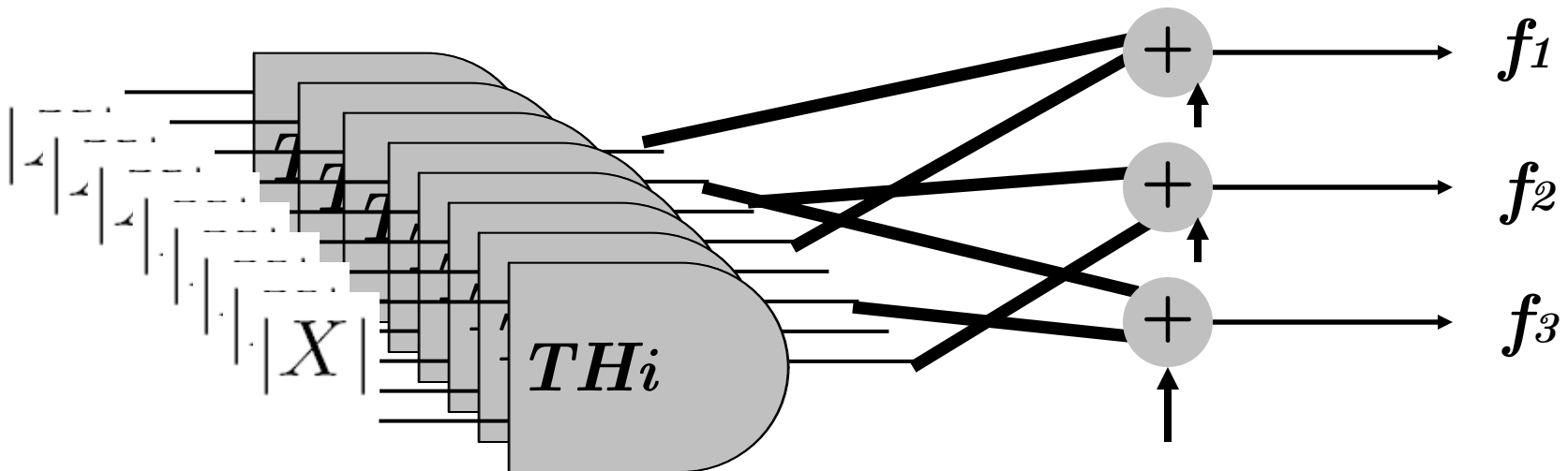
Q2: Why do we care about **symmetric** functions?
What about **general** Boolean functions?

Q3: For **LT-I** we proved that **depth** does not help in reducing the circuit size, does it help in the case of **LT-nl**?

Questions on LT Constructions

Q1: We understand how to implement a **single** symmetric function, what if we need to implement a circuit that computes a **number** of symmetric functions? Can we save in circuit size?

A1: YES - we can 'reuse' the first layer! By putting all required TH gates.



Questions on LT Constructions

Q1: we understand how to implement a **single** symmetric function, what if we need to implement a circuit that computes a **number** of symmetric functions?
Can we save in circuit size?

Q2: Why do we care about **symmetric** functions?
What about **general** Boolean functions?

Q3: For **LT-I** we proved that **depth** does not help in reducing the circuit size, does it help in the case of **LT-nl**?

Questions on LT Constructions



Q2: Why do we care about **symmetric** functions?

What about **general** Boolean functions?

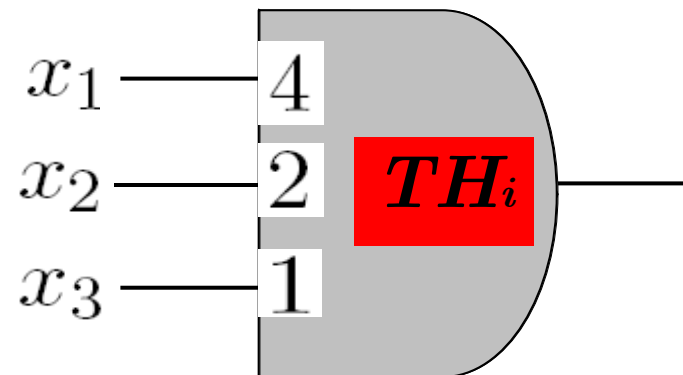
A2: A general Boolean function can be treated as a symmetric function by introducing **linear ordering**

$x_1x_2x_3$	$f(x_1, x_2, x_3)$	4 2 1
000	1	0
001	0	1
010	0	2
011	1	3
100	1	4
101	1	5
110	0	6
111	0	7

5 gates with DNF

4 gates with LT-I

3 gates with LT-nl



0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31
100000	32
etc.	etc

Questions on LT Constructions

Q1: We understand how to implement a **single** symmetric function, what if we need to implement a circuit that computes a **number** of symmetric functions?
Can we save in circuit size?

Q2: Why do we care about **symmetric** functions?
What about **general** Boolean functions?

Q3: For **LT-I** we proved that **depth does not help** in reducing the circuit size, **does it help** in the case of **LT-nl**?

Questions on LT Constructions



Q3: of for LT-I we proved that **depth does not help** in reducing the circuit size, **does it help** in the case of LT-nl?

A3: Yes!

Idea: compute the base-2 representation of $|X|$

$ X $	8421
0	0000
1	0001
5	0101
7	0111
8	1000
11	1011

$XOR(X)$

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20
10101	21
10110	22
10111	23
11000	24
11001	25
11010	26
11011	27
11100	28
11101	29
11110	30
11111	31
100000	32
etc.	etc.

Idea: compute the base-2 representation of $|X|$

Algorithm for computing the binary representation:

Input: an integer m

Assume: $2^{j-1} \leq m < 2^j$

$$\sum_{i=0}^{j-1} 2^i = 2^j - 1$$

Need to compute: j bits of the binary representation

Algorithm: $A \leftarrow m$

For $i = j - 1$ to 0 do:

{ if $A \geq 2^i$ then $a_i = 1$
else $a_i = 0$
 $A \leftarrow A - a_i 2^i$ }

Idea: compute the base-2 representation of $|X|$

Algorithm for computing the binary representation:

Input: an integer m

Assume: $2^{j-1} \leq m < 2^j$

$$\sum_{i=0}^{j-1} 2^i = 2^j - 1$$

Need to compute: j bits of the binary representation

Algorithm: $A \leftarrow 6$

Need 3 bits

For $i = j-1$ to 0 do:

{ if $6 \geq 4$ then $a_i = 1$
else $a_i = 0$

$$a_2 = 1$$

$A \leftarrow A - 2^i$ }

Idea: compute the base-2 representation of $|X|$

Algorithm for computing the binary representation:

Input: an integer m

Assume: $2^{j-1} \leq m < 2^j$

$$\sum_{i=0}^{j-1} 2^i = 2^j - 1$$

Need to compute: j bits of the binary representation

Algorithm: $A \leftarrow 2$

Need 3 bits

For $i = j-1$ to 0 do:

{ if $2 \geq 2$ then $a_i = 1$
else $a_i = 0$

$A \leftarrow A - 2^i$ }

$$a_2 = 1$$

$$a_1 = 1$$

Idea: compute the base-2 representation of $|X|$

Algorithm for computing the binary representation:

Input: an integer m

Assume: $2^{j-1} \leq m < 2^j$

$$\sum_{i=0}^{j-1} 2^i = 2^j - 1$$

Need to compute: j bits of the binary representation

Algorithm: $A \leftarrow 0$

Need 3 bits

For $i = j-1$ to 0 do:

{ if $m \geq 2^i$ then $a_i = 1$
 else $a_i = 0$
 $A \leftarrow A + 2^i$ }

$$a_2 = 1$$

$$a_1 = 1$$

$$a_0 = 0$$

Idea: compute the base-2 representation of $|X|$

Algorithm for computing the binary representation:

Input: an integer m

Assume: $2^{j-1} \leq m < 2^j$

$$\sum_{i=0}^{j-1} 2^i = 2^j - 1$$

Need to compute: j bits of the binary representation

Algorithm: $A \leftarrow m$

For $i = j - 1$ to 0 do:

{ if $A \geq 2^i$ then $a_i = 1$
else $a_i = 0$
 $A \leftarrow A - a_i 2^i$ }

$$a_2 = 1$$

$$a_1 = 1$$

$$a_0 = 0$$

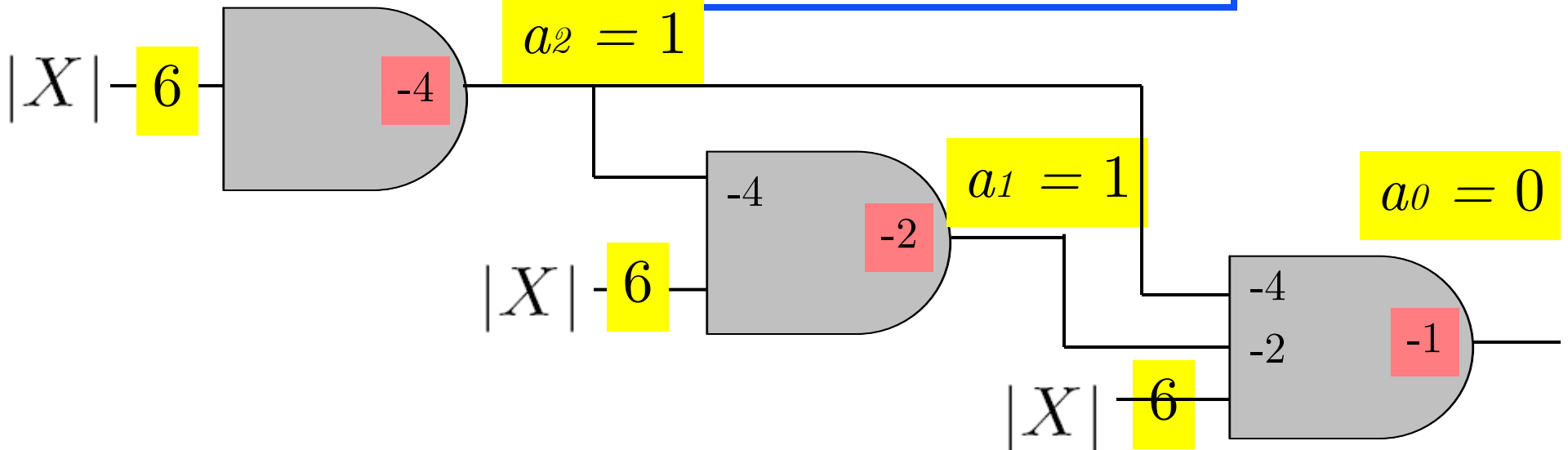
The Kautz Construction 1961

Implement the algorithm for computing the binary representation using LT gates

Algorithm: $A \leftarrow m$

For $i = j - 1$ to 0 do:

{ if $A \geq 2^i$ then $a_i = 1$
else $a_i = 0$
 $A \leftarrow A - a_i 2^i$ }



LT Constructions for Symmetric Functions

circuit kind	size
LT-I	$n + 1$
LT-nl, d-2	$\left\lfloor \frac{n}{2} \right\rfloor + 1$
LT-nl -XOR	$\lceil \log_2(n + 1) \rceil$



optimal

LT-nl Construction

Minnick 1961

Linear-Input Logic*

ROBERT C. MINNICK[†], MEMBER, IRE

Summary—Techniques are developed for the logical design of magnetic core circuits to produce arbitrary single-output combinational switching functions. The approach is based on the relationship of a single magnetic core circuit to a linearly separable switching function. A synthesis procedure is developed which uses a pair of logical primitives, AND with NOT and OR with NOT, which are similar to the STROKE primitive and its inverse. Procedures are developed for the synthesis of symmetric functions which require no more than the integral part of $(n+3)/2$ cores, approximately half the number used in previously published procedures. The synthesis of arbitrary switching circuits is treated as a linear programming problem, and a table of all four-variable circuits is presented in which no circuit requires more than three cores.

First construction

[5] S. Muroga, "Logical Elements on Majority Decision Principle and Complexity of Their Circuit," presented at Internatl. Conf. on Information Processing, Paris, France; June 15–20, 1959.

Minnick's construction

[6] R. C. Minnick, "Synthesis of Linear-Input Logic by the Simplex Method," presented at the Sixth Annual Symp. on Computers and Data Processing of the Denver Res. Inst., Denver, Colo.; July 30, 1959.

LT-nl Construction

Minnick 1961

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS

Linear-Input Logic*

ROBERT C. MINNICK†, MEMBER, IRE

* Received by the PGEC, August 17, 1960; revised manuscript received, October 3, 1960. This paper is the result of work done while the author was employed by Burroughs Corp., ElectroData Div., Pasadena, Calif. Portions of this paper were delivered at the Sixth Annual Symp. on Computers and Data Processing of the Denver Res. Inst., Denver, Colo.; July, 1959.

† Stanford Res. Inst., Menlo Park, Calif.

The Kautz Construction 1961

1961

IRE TRANSACTIONS ON ELECTRONIC COMPUTERS

The Realization of Symmetric Switching Functions with Linear-Input Logical Elements*

WILLIAM H. KAUTZ[†], MEMBER, IRE

Previously published solutions for the case of the parity function have required from $1 + \lceil n/2 \rceil$ to $1 + n$ elements.^{1,4} The improved solution presented below makes possible simple parity-checking and error-correction circuits for applications to the recording and transmission of digital data. A 7-input parity gate, for example, requires only 3 linear-input elements (Fig. 2); a gate with up to 15 inputs requires 4 elements, etc.

The Kautz Construction

1961

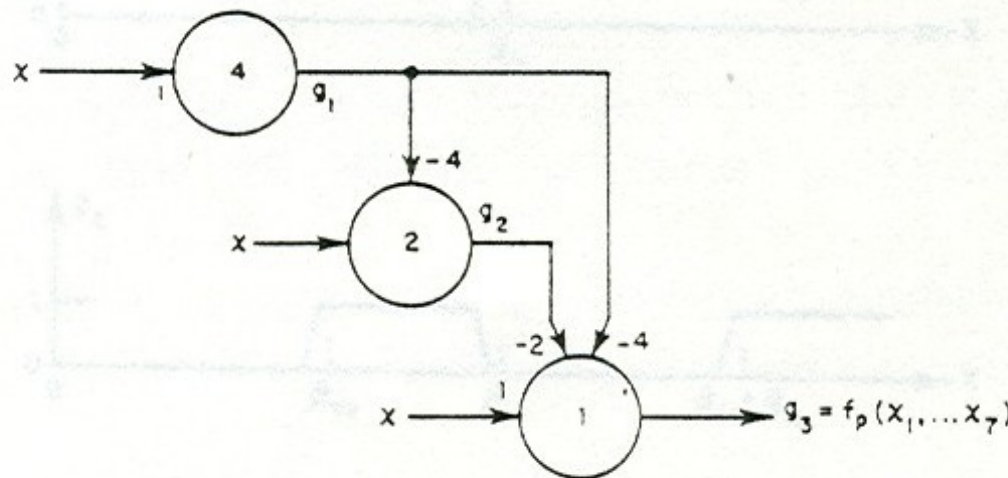
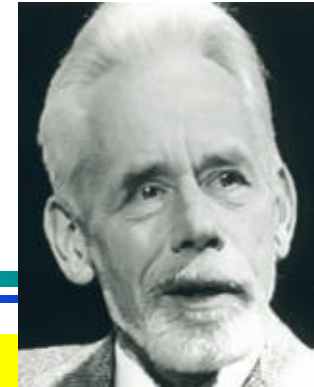


Fig. 2—Minimal linear-input network for the 7-variable parity function.

ACKNOWLEDGMENT

The author is indebted to **R. Minnick** of Stanford Research Institute, whose work stimulated his serious interest in the subject area of this paper, and with whom many profitable technical discussions have taken place.

Kautz



William H. Kautz (Sc.D., M.I.T).
was Staff Scientist at Stanford Research Institute,
where for 35 years he conducted scientific and technical research
in computer science and various other fields.

He then founded and directed the Center for Applied Intuition,
which carried out research on intuition and its applications.

He lives in Prague, Czech Republic, and Tucson, Arizona.

