Even and Odd Parity Generators and Checkers

SHANTHARAM P – [CB.SC.U4CSE23146]
SUTHEKSHAN ADEIT AYYAKANNU – [CB.SC.U4CSE23159]

Even and Odd Parity Generators and Checkers

1. Introduction

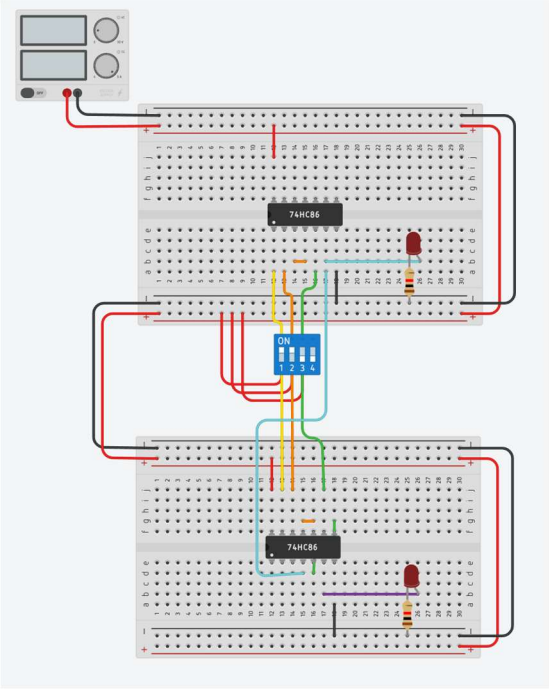
In digital communication, error detection plays a crucial role in ensuring the integrity of data transmission. One common method for error detection is the use of parity bits. Parity is a simple technique that involves adding an extra bit to a binary message to make the number of 1s either even (even parity) or odd (odd parity).

Parity generators and checkers are essential components in error detection mechanisms used in data transmission. This report focuses on the design and simulation of even and odd parity generators and checkers.

2. Circuit Design and Explanation

Even Parity Generator and Checker Circuit Design

The even parity generator circuit adds a parity bit to ensure the number of 1s in the message is even. The even parity checker checks whether the number of 1s in the circuit is even after the addition of the parity bit. The even parity generator and checker use XOR gates but ensure an even number of 1s.



The truth table for the even parity generator is as follows,

A	B	C	P(Parity)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Drawing K-Map,

A\BC	BC=00	BC=01	BC=11	BC=10
A=0	0	1	0	1
A=1	1	0	1	0

From the K-MAP we get the SOP,

P=AB'C'+A'BC'+A'B'C+ABC

This is exactly the equivalent of P=A⊕B⊕C and therefore the minimized SOP.

The even parity checker also follows the same ex-or logic,i.e. Output is X = A⊕B⊕C⊕P.

Odd Parity Generator and Checker Circuit Design

The odd parity generator circuit adds a parity bit to ensure the number of 1s in the message is odd. The odd parity checker checks whether the number of 1's in the circuit is odd after the addition of parity bit.
The odd parity generator and checker uses XOR gates but ensures an odd number of 1s. The circuit is similar to the even parity generator and checker but the output is inverted.

The truth table for the odd parity generator is as follows,

A	B	C	P(Parity)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Drawing K-Map,

A\BC	BC=00	BC=01	BC=11	BC=10
A=0	1	0	1	0
A=1	0	1	0	1

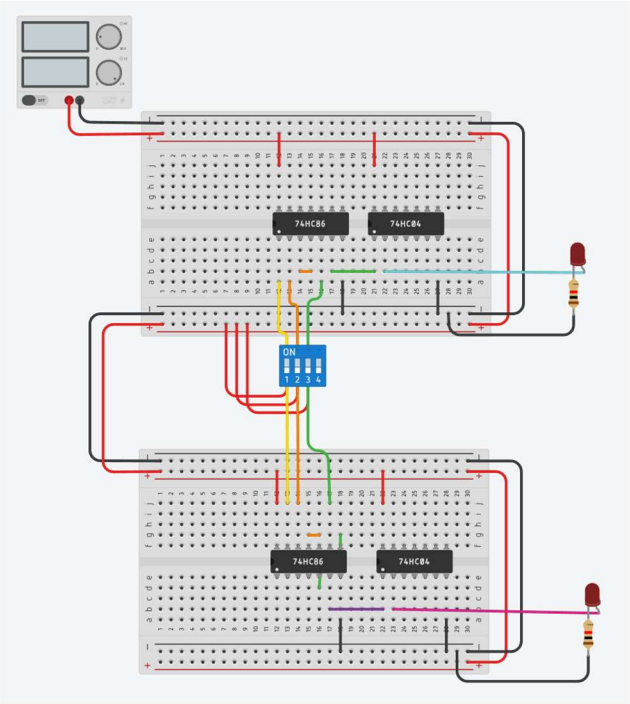
From K-Map we got the SOP,

P= A'BC +AB'C + ABC' + A'B'C'

This expression is the inverted form of the experession we got in the even parity generator.

Hence we conclude that the minimized SOP is, P = (A⊕B⊕C)'

The odd parity checker also follows the same logic, the output X= (A⊕B⊕C⊕P)'

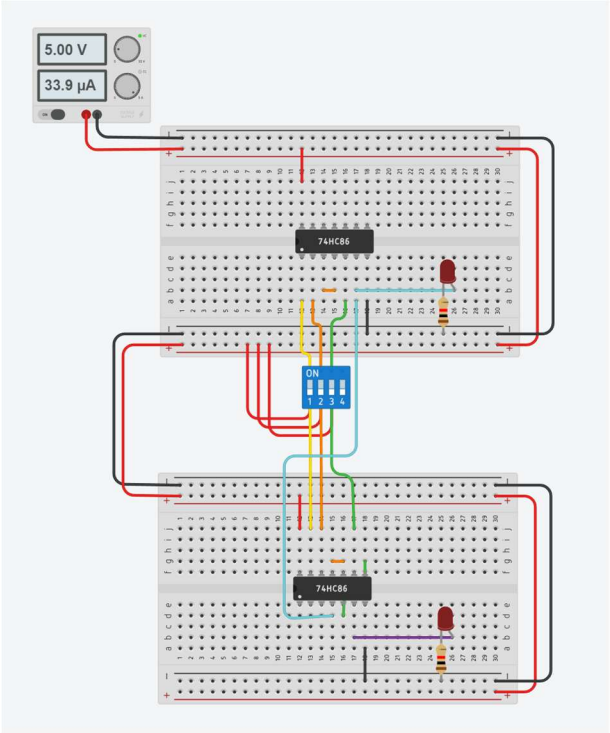


3. Simulation Results

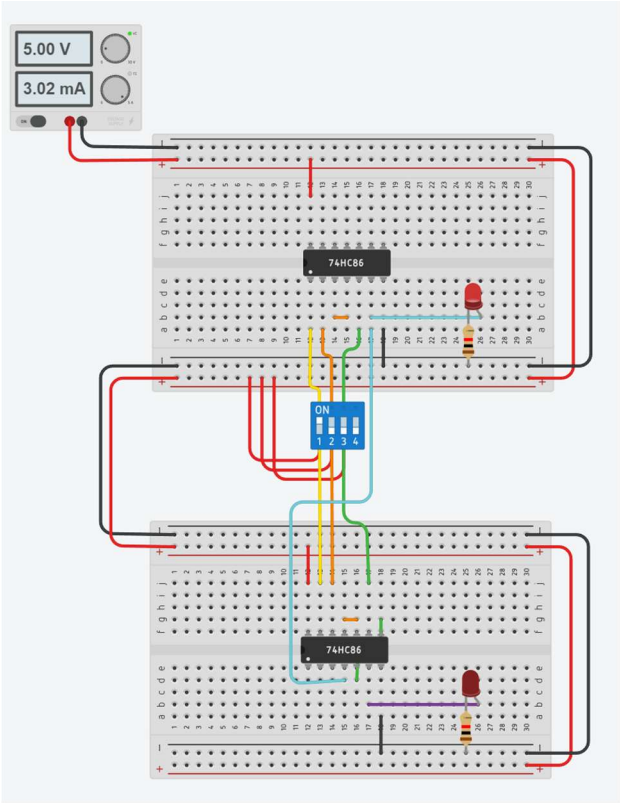
The following section presents the simulation results for the designed even and odd parity generators and checkers.
The results are obtained using a digital circuit simulation tool.

Even Parity Generator and Checker Results

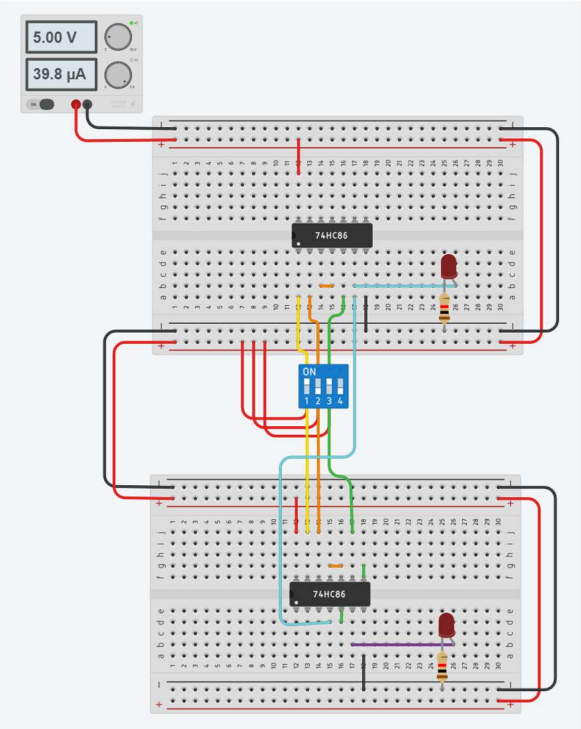
When no input is HIGH, the parity bit is also not generated.



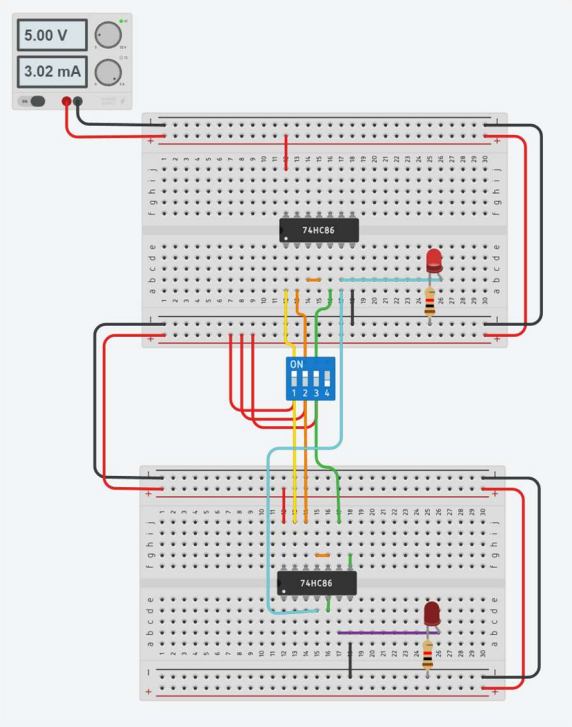
When any one of the inputs is HIGH, the parity bit is generated i.e. HIGH is generated.



When any two of the inputs is HIGH, the parity bit is not generated i.e. LOW



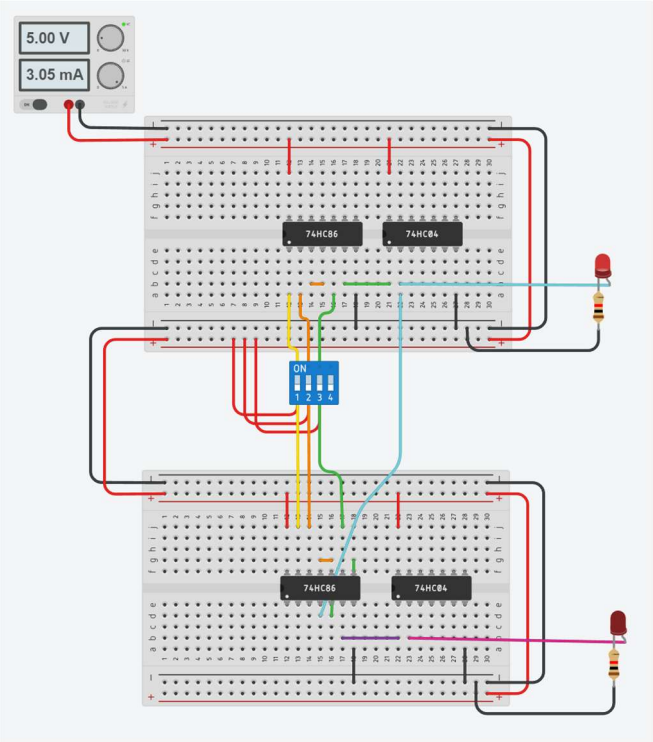
When all three of the input is HIGH, the parity bit is generated i.e. HIGH is generated



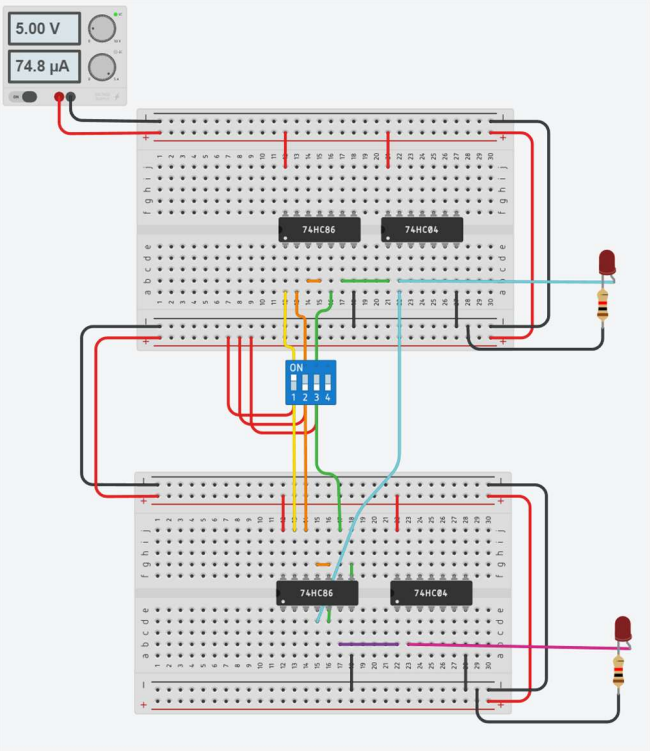
In all of these cases, the even parity checker won't produce HIGH since the total HIGH's will be always even.

Odd Parity Generator and Checker Results

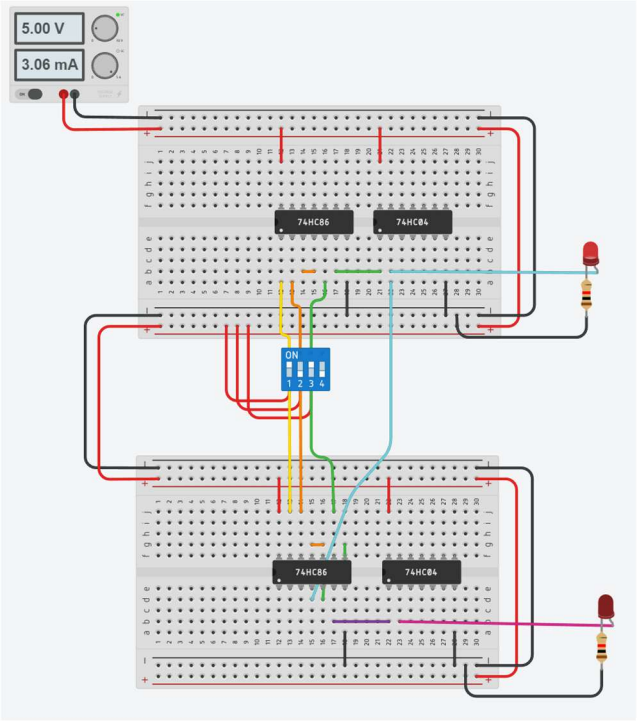
When no input is HIGH, the parity bit is generated i.e. HIGH



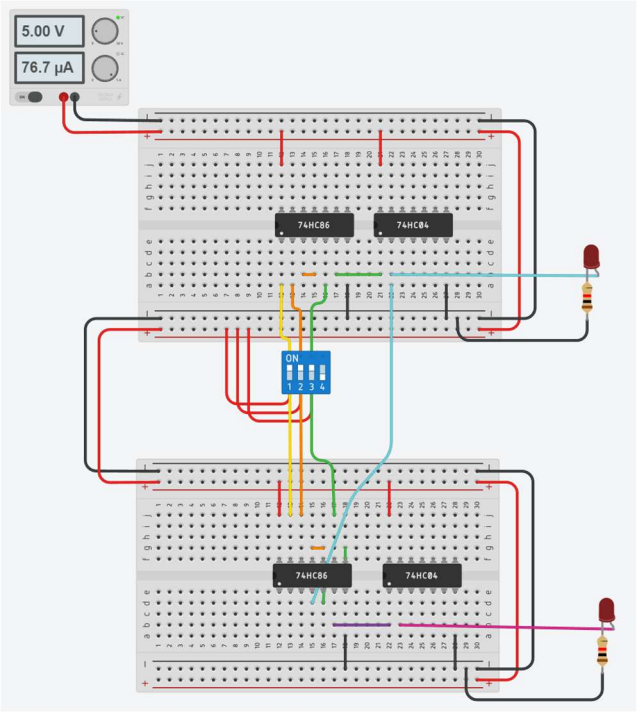
When any one of the inputs is HIGH, parity bit is not generated i.e LOW



When any two of the inputs are HIGH, the parity bit is generated i.e. HIGH



When all three of the inputs are HIGH, the parity bit is not generated i.e. LOW



In all of these cases, the odd parity checker won't produce HIGH since the total HIGH's will be always odd.

4. Analysis and Discussion

Based on the simulation results, the following analysis discusses the performance of the parity generators and checkers.

First, the even parity generator receives three bits (for example), checks the input bits, and verifies whether the number of HIGH inputs is even. If the number of HIGH inputs is not even, the parity generator outputs a HIGH parity bit; otherwise, the parity bit is LOW. The even parity checker takes in four bits as input, which includes the three original bits and the parity bit. These inputs are passed through an XOR gate to verify whether the number of HIGH inputs is even. When you pass the three original inputs along with the parity bit, the total number of HIGHs will always be even, ensuring correct error detection.

For odd parity, the process is similar but with reversed logic. The odd parity generator also receives three input bits and checks whether the number of HIGH inputs is odd. If the number of HIGH inputs is not odd, the generator produces a HIGH parity bit; if it is already odd, the parity bit is LOW. The odd parity checker accepts four bits, including the three original bits and the parity bit, and passes them through an XOR gate. This checks whether the total number of HIGHs is odd. By combining the three inputs with the parity bit, the total number of HIGHs will always be odd, ensuring proper error detection for odd parity systems.

5. Conclusion

Parity generators and checkers provide a straightforward yet powerful solution for error detection in digital communication systems. The circuits analyzed in this report highlight the effectiveness of both even and odd parity in maintaining the integrity of transmitted data. Due to their simplicity and low resource requirements, parity-based error detection continues to play a crucial role in ensuring reliable communication across various digital systems.