

# ELECTRONICS HUB

PROJECTS | TUTORIALS | COURSES | KITS

[HOME](#)[PROJECTS](#)[MINI PROJECTS](#)[FREE CIRCUITS](#)[TUTORIALS](#)[SYMBOLS](#)[DIY](#)[PROJECT KITS](#)[COURSES](#)[CONTACT US](#)

YOU ARE HERE: [HOME](#) / [COMBINATIONAL LOGIC CIRUCITS](#) / [PARITY GENERATOR AND PARITY CHECK](#)

## Parity Generator and Parity Check

JULY 7, 2015 BY [ADMINISTRATOR](#) — [8 COMMENTS](#)

### Table of Contents



1. What is Parity Bit?
2. Parity generator and checker
3. Parity Generator
  - 3.1. Even Parity Generator
  - 3.2. Odd Parity Generator
4. Parity Check
  - 4.1. Even Parity Checker
  - 4.2. Odd Parity Checker
5. Parity Generator/Checker ICs

### What is Parity Bit?

The parity generating technique is one of the most widely used error detection techniques for the data transmission. In digital systems, when binary data is transmitted and processed , data may be subjected to noise so that such noise can alter 0s (of data bits) to 1s and 1s to 0s.

Hence, **parity bit** is added to the word containing data in order to make number of 1s either even or odd. Thus it is used to detect errors, during the transmission of binary data. The message containing the data bits along with parity bit is transmitted from transmitter node to receiver node.

At the receiving end, the number of 1s in the message is counted and if it doesn't match with the transmitted one, then it means there is an error in the data.

[Back to top](#)

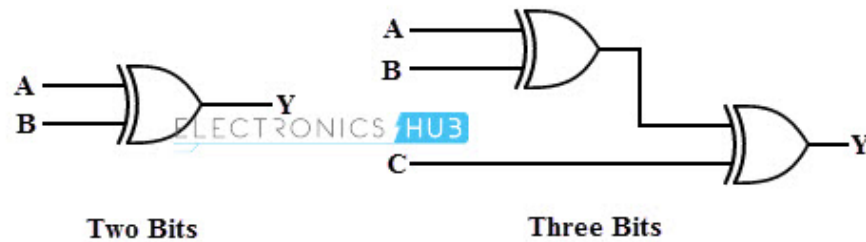
## Parity generator and checker

A parity generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called parity checker. A combined circuit or devices of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data word.

The sum of the data bits and parity bits can be even or odd. In even parity, the added parity bit will make the total number of 1s an even amount whereas in odd parity the added parity bit will make the total number of 1s odd amount.

The basic principle involved in the implementation of parity circuits is that sum of odd number of 1s is always 1 and sum of even number of 1s is always zero. Such error detecting and correction can be implemented by using Ex-OR gates (since Ex-OR gate produces zero output when there are even number of inputs).

To produce two bits sum, one Ex-OR gate is sufficient whereas for adding three bits two Ex-OR gates are required as shown in below figure.

[Back to top](#)

## Parity Generator

It is combinational circuit that accepts an  $n-1$  bit stream data and generates the additional bit that is to be transmitted with the bit stream. This additional or extra bit is termed as a parity bit.

In **even parity** bit scheme, the parity bit is '0' if there are **even number of 1s** in the data stream and the parity bit is '1' if there are **odd number of 1s** in the data stream.

In **odd parity** bit scheme, the parity bit is '1' if there are **even number of 1s** in the data stream and the parity bit is '0' if there are **odd number of 1s** in the data stream. Let us discuss both even and odd parity generators.

## Even Parity Generator

Let us assume that a 3-bit message is to be transmitted with an even parity bit. Let the three inputs A, B and C are applied to the circuits and output bit is the parity bit P. The total number of 1s must be even, to generate the even parity bit P.

The figure below shows the truth table of even parity generator in which 1 is placed as parity bit in order to make all 1s as even when the number of 1s in the truth table is odd.

3-bit message			Even parity bit generator (P)
A	B	C	Y
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

The K-map simplification for 3-bit message even parity generator is

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

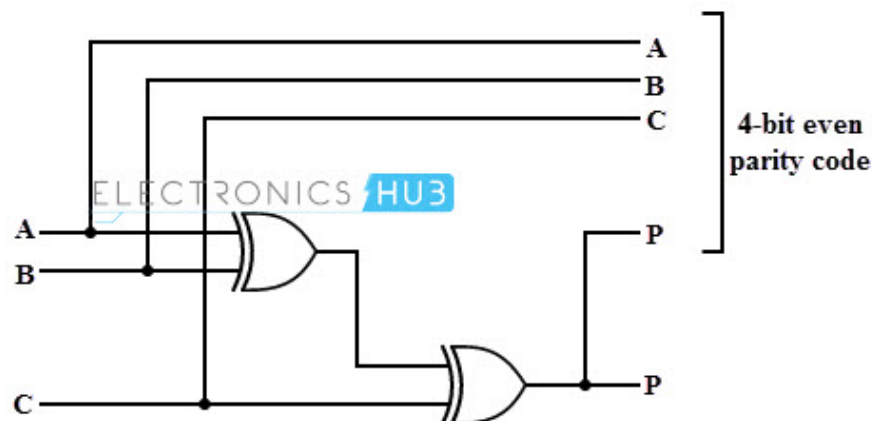
From the above truth table, the simplified expression of the parity bit can be written as

$$\begin{aligned}
 P &= \bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} + A B C \\
 &= \bar{A} (\bar{B} C + B \bar{C}) + A (\bar{B} \bar{C} + B C) \\
 &= \bar{A} (B \oplus C) + A (\overline{B \oplus C}) \\
 P &= A \oplus B \oplus C
 \end{aligned}$$

The above expression can be implemented by using two Ex-OR gates. The logic diagram of even parity generator with two Ex – OR gates is shown below. The three bit message along

with the parity generated by this circuit which is transmitted to the receiving end where parity checker circuit checks whether any error is present or not.

To generate the even parity bit for a 4-bit data, three Ex-OR gates are required to add the 4-bits and their sum will be the parity bit.



[Back to top](#)

## Odd Parity Generator

Let us consider that the 3-bit data is to be transmitted with an odd parity bit. The three inputs are A, B and C and P is the output parity bit. The total number of bits must be odd in order to generate the odd parity bit.

In the given truth table below, 1 is placed in the parity bit in order to make the total number of bits odd when the total number of 1s in the truth table is even.

3-bit message			Odd parity bit generator (P)
A	B	C	Y
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

The truth table of the odd parity generator can be simplified by using K-map as

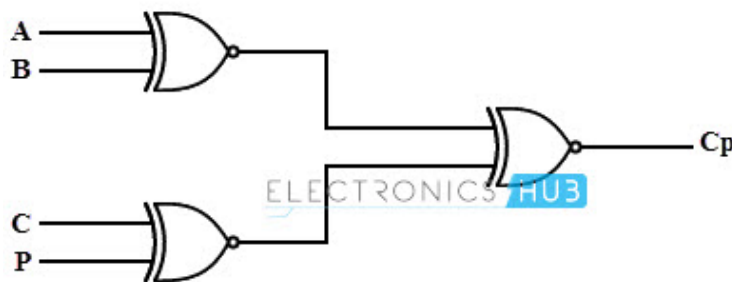
		BC			
		00	01	11	10
A	00	0 <b>1</b>	1 <b>0</b>	3 <b>1</b>	2 <b>0</b>
	01	4 <b>0</b>	5 <b>1</b>	7 <b>0</b>	6 <b>1</b>

The output parity bit expression for this generator circuit is obtained as

$$P = A \oplus B \text{ Ex-NOR } C$$

The above Boolean expression can be implemented by using one Ex-OR gate and one Ex-NOR gate in order to design a 3-bit odd parity generator.

The logic circuit of this generator is shown in below figure , in which . two inputs are applied at one Ex-OR gate, and this Ex-OR output and third input is applied to the Ex-NOR gate , to produce the odd parity bit. It is also possible to design this circuit by using two Ex-OR gates and one NOT gate.



[Back to top](#)

## Parity Check

It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the

transmission end. When this circuit is used as even parity checker, the number of input bits must always be even.

When a parity error occurs, the 'sum even' output goes low and 'sum odd' output goes high. If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the 'sum odd' output goes low and 'sum even' output goes high.

## Even Parity Checker

Consider that three input message along with even parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit which checks the possibility of error on the data. Since the data is transmitted with even parity, four bits received at circuit must have an even number of 1s.

If any error occurs, the received message consists of odd number of 1s. The output of the parity checker is denoted by PEC (parity error check).

The below table shows the truth table for the even parity checker in which  $PEC = 1$  if the error occurs, i.e., the four bits received have odd number of 1s and  $PEC = 0$  if no error occurs, i.e., if the 4-bit message has even number of 1s.

The above truth table can be simplified using K-map as shown below.

The above logic expression for the even parity checker can be implemented by using three Ex-OR gates as shown in figure. If the received message consists of five bits, then one more Ex-OR gate is required for the even parity checking.

[Back to top](#)

## Odd Parity Checker

Consider that a three bit message along with odd parity bit is transmitted at the transmitting end. Odd parity checker circuit receives these 4 bits and checks whether any error are present in the data.

If the total number of 1s in the data is odd, then it indicates no error, whereas if the total number of 1s is even then it indicates the error since the data is transmitted with odd parity at transmitting end.

The below figure shows the truth table for odd parity generator where **PEC =1** if the 4-bit message received consists of **even number of 1s** (hence the error occurred) and **PEC= 0** if the message contains **odd number of 1s** (that means no error).

The expression for the PEC in the above truth table can be simplified by K-map as shown below.

After simplification, the final expression for the PEC is obtained as

$$PEC = (A \text{ Ex-NOR } B) \text{ Ex-NOR } (C \text{ Ex-NOR } D)$$

The expression for the odd parity checker can be designed by using three Ex-NOR gates as shown below.

[Back to top](#)

## Parity Generator/Checker ICs

There are different types of parity generator /checker ICs are available with different input configurations such as 5-bit, 4-bit, 9-bit, 12-bit, etc. A most commonly used and standard type of parity generator/checker IC is 74180.

It is a 9-bit parity generator or checker used to detect errors in high speed data transmission or data retrieval systems. The figure below shows the pin diagram of 74180 IC.

This IC can be used to generate a 9-bit odd or even parity code or it can be used to check for odd or even parity in a 9-bit code (8 data bits and one parity bit).

This IC consists of eight parity inputs from A through H and two cascading inputs. There are two outputs even sum and odd sum. In implementing generator or checker circuits, unused parity bits must be tied to logic zero and the cascading inputs must not be equal.



FILED UNDER: [COMBINATIONAL LOGIC CIRUCITS](#)

## Comments



**Trupalsinh says**

MAY 8, 2016 AT 11:01 PM

Superb..!!!! All informations are covered....

[Reply](#)



**aditi says**

JULY 21, 2016 AT 1:01 AM

awesome...

[Reply](#)

**naveen reddy says**

SEPTEMBER 12, 2016 AT 2:10 PM

awesom nd tq

[Reply](#)**swali khan says**

SEPTEMBER 15, 2016 AT 11:34 AM

it is covered all topic.it is very help full for me.

[Reply](#)**vishnu acharya says**

SEPTEMBER 30, 2016 AT 5:55 AM

nice and usefull

[Reply](#)**simi says**

NOVEMBER 28, 2016 AT 12:29 AM

Is exnor or exor or both In the odd parity generator. It is confusing with sentence and figure

[Reply](#)

**tarun says**

APRIL 3, 2017 AT 2:23 AM

tell some important applications

[Reply](#)**Anjali Rai says**

MARCH 13, 2018 AT 5:53 AM

Thank you superb knowledge shared

[Reply](#)

## Leave a Reply

Your email address will not be published. Required fields are marked \*

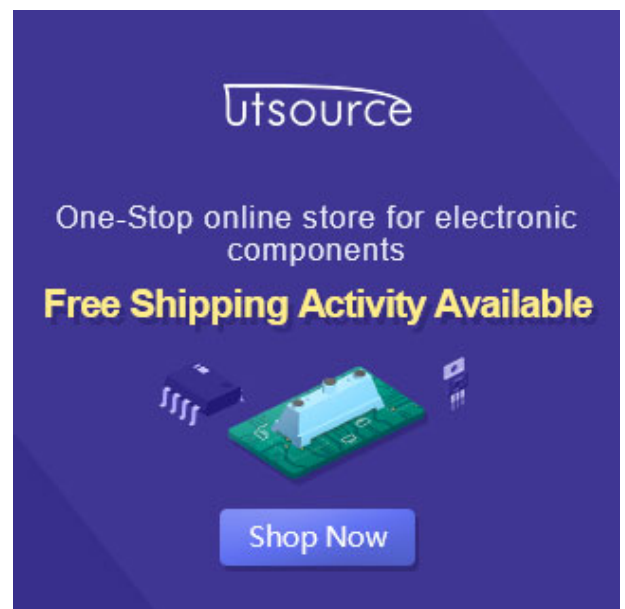
Comment

Name \*

Email \*

Website

POST COMMENT



Utsource

One-Stop online store for electronic components

**Free Shipping Activity Available**

Shop Now

The advertisement banner for Utsource features a dark blue background. At the top, the 'Utsource' logo is displayed in white. Below the logo, the text 'One-Stop online store for electronic components' is written in white. A prominent yellow text overlay reads 'Free Shipping Activity Available'. In the center, there is a 3D illustration of electronic components: a blue integrated circuit, a green printed circuit board with a white component, and a small blue component. At the bottom, a blue button with the text 'Shop Now' in white is visible.



---

**Suggest A Topic [or] Project**

---

---

---

PROJECTS BY CATEGORY

---

**Arduino Projects (200+)**  
**Electronics Projects (250+)**  
**Mini Project Circuits (160+)**  
**Mini Project Ideas (150+)**  
**ECE Projects (150+)**  
**EEE Projects (150+)**  
**8051 Projects (110+)**  
**Raspberry Pi Projects (101+)**  
**Electrical Project Ideas (100+)**  
**Embedded Projects (100+)**  
**Latest Electronics Ideas (100+)**  
**Microcontroller Mini Projects (100+)**  
**Robotics Projects (100+)**  
**VLSI Projects (100+)**  
**Solar Projects (100+)**  
**IOT Projects (100+)**

---

**Communication Projects (70+)**  
**LED Projects (70+)**  
**Power Electronics Projects (60+)**  
**RFID Projects (60+)**  
**Home Automation Projects (50+)**  
**Matlab Projects (50+)**  
**EIE Projects (50+)**  
**Wireless Projects (50+)**  
**LabView Projects (45+)**  
**Zigbee Projects (45+)**  
**GSM Projects (40+)**  
**555 Timer Circuits (40+)**  
**Sensor Projects (40+)**  
**ARM Projects (60+)**  
**DTMF Projects (30+)**  
**PIC Projects (30+)**  
**Electrical Mini Projects (25)**

---

KITS

---

[Best Drone Kits \[12\]](#)  
[3D Printer Kits \[12\]](#)  
[Best Robot Vacuum Clears \[14\]](#)

GENERAL

[Tutorials](#)  
[Symbols](#)  
[Courses](#)  
[Calculator](#)  
[Contact](#)

PROJECTS

[Electrical](#)  
[Electronics](#)  
[Embedded](#)  
[Power](#)  
[Robotics](#)  
[ARM](#)  
[IOT](#)

PROJECTS

[Mini projects](#)  
[Microcontroller](#)  
[Arduino](#)  
[Solar](#)  
[Free circuits](#)  
[Home Automation](#)  
[Seminar Topics](#)  
[Electronics Questions](#)

TUTORIALS

[Capacitors](#)  
[Resistors](#)  
[Filters](#)  
[Diodes](#)  
[Transistors](#)

TUTORIALS

[Amplifiers](#)  
[IO Devices](#)  
[Thyristors](#)  
[DC Circuits](#)  
[Number System](#)

FOLLOW US

[Instagram](#)  
[Youtube](#)  
[Facebook](#)  
[Google Plus](#)  
[Twitter](#)