

[Home](#) [Computer Network](#) [DBMS](#) [DS](#) [DAA](#) [Operating Systems](#) [Computer Fundamentals](#) [Java](#) [Python](#) [HTML](#) [Selenium](#)

[↑ SCROLL TO TOP](#)

# Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:

- **Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.
- **Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors.

A single additional bit can detect the error, but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.



javaTpoint

## STACK IN DS

C PROGRAM TO IMPLEMENT  
STACK

Suppose  $r$  is the number of redundant bits and  $d$  is the total number of the data bits. The number of redundant bits  $r$  can be calculated by using the formula:

$$2^r \geq d + r + 1$$

The value of  $r$  is calculated by using the above formula. For example, if the value of  $d$  is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

## Hamming Code

**Parity bits:** The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

**Even parity:** To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

**Odd Parity:** To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

### Algorithm of Hamming code:

- An information of ' $d$ ' bits are added to the redundant bits ' $r$ ' to form  $d+r$ .
- The location of each of the  $(d+r)$  digits is assigned a decimal value.
- The ' $r$ ' bits are placed in the positions  $1, 2, \dots, 2^{k-1}$ .

- At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

## Relationship b/w Error position & binary number.

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

Suppose the original data is 1010 which is to be sent.

**Total number of data bits 'd' = 4**

**Number of redundant bits r :  $2^r \geq d+r+1$**

$$2^r \geq 4 + r + 1$$

Therefore, the value of  $r$  is 3 that satisfies the above relation.

**Total number of bits =  $d + r = 4 + 3 = 7$ ;**

## Determining the position of the redundant bits

The number of redundant bits is 3. The three bits are represented by  $r_1$ ,  $r_2$ ,  $r_4$ . The position of the redundant bits is calculated with corresponds to the raised power of 2. Therefore, their corresponding positions are **1,  $2^1$ ,  $2^2$** .

The position of  $r_1 = 1$

The position of  $r_2 = 2$

The position of  $r_4 = 4$

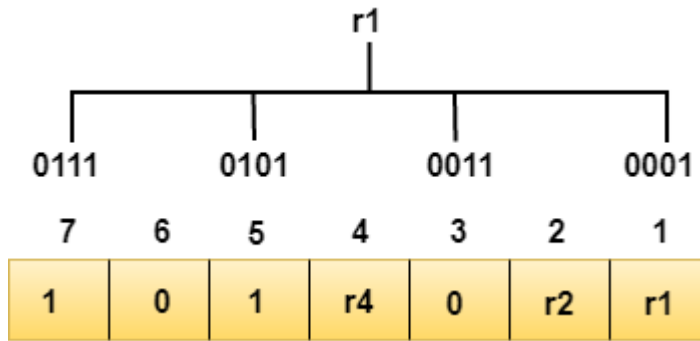
Representation of Data on the addition of parity bits:

7	6	5	4	3	2	1
1	0	1	$r_4$	0	$r_2$	$r_1$

## Determining the Parity bits

### Determining the $r_1$ bit

The  $r_1$  bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



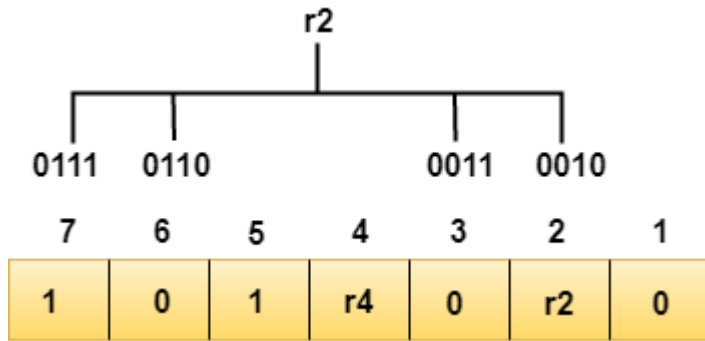
We observe from the above figure that the bit positions that includes 1 in the first position are 1, 3, 5, 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r1 is **even, therefore, the value of the r1 bit is 0.**

## Determining r2 bit

The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.

₹9

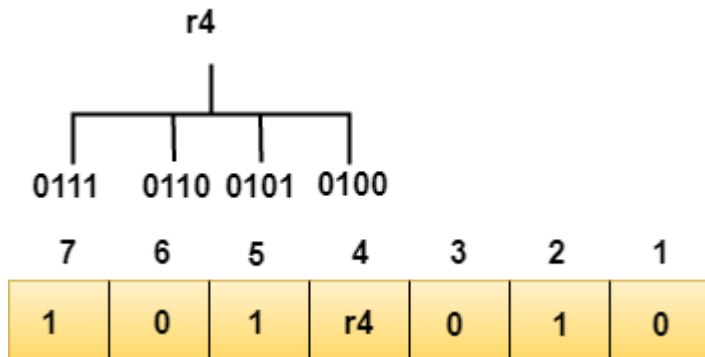
₹9



We observe from the above figure that the bit positions that includes 1 in the second position are **2, 3, 6, 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r2 is **odd, therefore, the value of the r2 bit is 1**.

### Determining r4 bit

The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



We observe from the above figure that the bit positions that includes 1 in the third position are **4, 5, 6, 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r4 is **even, therefore, the value of the r4 bit is 0**.

**Data transferred is given below:**

7	6	5	4	3	2	1
1	0	1	0	0	1	0

Suppose the 4<sup>th</sup> bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

## R1 bit

The bit positions of the r1 bit are 1,3,5,7

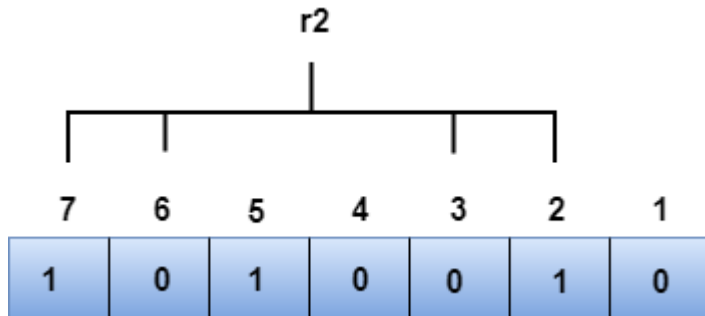
			r1			
			├───┬───┬───┬───┤			
7	6	5	4	3	2	1
1	0	1	1	0	1	0

We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

## R2 bit

The bit positions of r2 bit are 2,3,6,7.

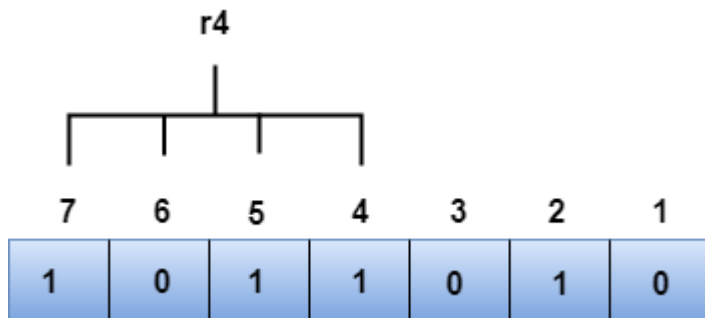




We observe from the above figure that the binary representation of r2 is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the r2 bit is an even number. Therefore, the value of r2 is 0.

### R4 bit

The bit positions of r4 bit are 4,5,6,7.



We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

- The binary representation of redundant bits, i.e.,  $r_4r_2r_1$  is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4<sup>th</sup> bit position. The bit value must be changed from 1 to 0 to correct the error.

← Prev

Next →

 Youtube For Videos Join Our Youtube Channel: [Join Now](#)

## Feedback











- Send your Feedback to [feedback@javatpoint.com](mailto:feedback@javatpoint.com)

## Help Others, Please Share





## Learn Latest Tutorials

 Splunk tutorial Splunk	 SPSS tutorial SPSS	 Swagger tutorial Swagger	 T-SQL tutorial Transact-SQL	 Tumblr tutorial Tumblr
 React tutorial ReactJS	 Regex tutorial Regex	 Reinforcement learning tutorial Reinforcement Learning	 R Programming tutorial R Programming	 RxJS tutorial RxJS



React Native  
tutorial

React Native



Python Design  
Patterns

Python Design  
Patterns



Python Pillow  
tutorial

Python Pillow



Python Turtle  
tutorial

Python Turtle



Keras tutorial  
Keras

Keras

## Preparation



Aptitude

Aptitude



Logical  
Reasoning

Reasoning



Verbal Ability

Verbal Ability



Interview  
Questions

Interview Questions



Company  
Interview  
Questions

Company Questions

## Trending Technologies



Artificial  
Intelligence

Artificial  
Intelligence



AWS Tutorial

AWS



Selenium  
tutorial

Selenium



Cloud  
Computing

Cloud Computing



Hadoop tutorial

Hadoop



ReactJS  
Tutorial

ReactJS



Data Science  
Tutorial

Data Science



Angular 7  
Tutorial

Angular 7



Blockchain  
Tutorial

Blockchain



Git Tutorial

Git



Machine  
Learning Tutorial  
Machine Learning



DevOps  
Tutorial  
DevOps

## B.Tech / MCA



DBMS tutorial  
DBMS



Data Structures  
tutorial  
Data Structures



DAA tutorial  
DAA



Operating  
System  
Operating System



Computer  
Network tutorial  
Computer Network



Compiler  
Design tutorial  
Compiler Design



Computer  
Organization and  
Architecture
















Discrete  
Mathematics  
Tutorial



Ethical Hacking  
Ethical Hacking



Computer  
Graphics Tutorial  
Computer Graphics

	Computer Organization	Discrete Mathematics		
 Software Engineering	 html tutorial Web Technology	 Cyber Security tutorial Cyber Security	 Automata Tutorial Automata	 C Language tutorial C Programming
 C++ tutorial C++	 Java tutorial Java	 .Net Framework tutorial .Net	 Python tutorial Python	 List of Programs Programs
 Control Systems tutorial Control System	 Data Mining Tutorial Data Mining	 Data Warehouse Tutorial Data Warehouse		

