# EXPERIMENT 3

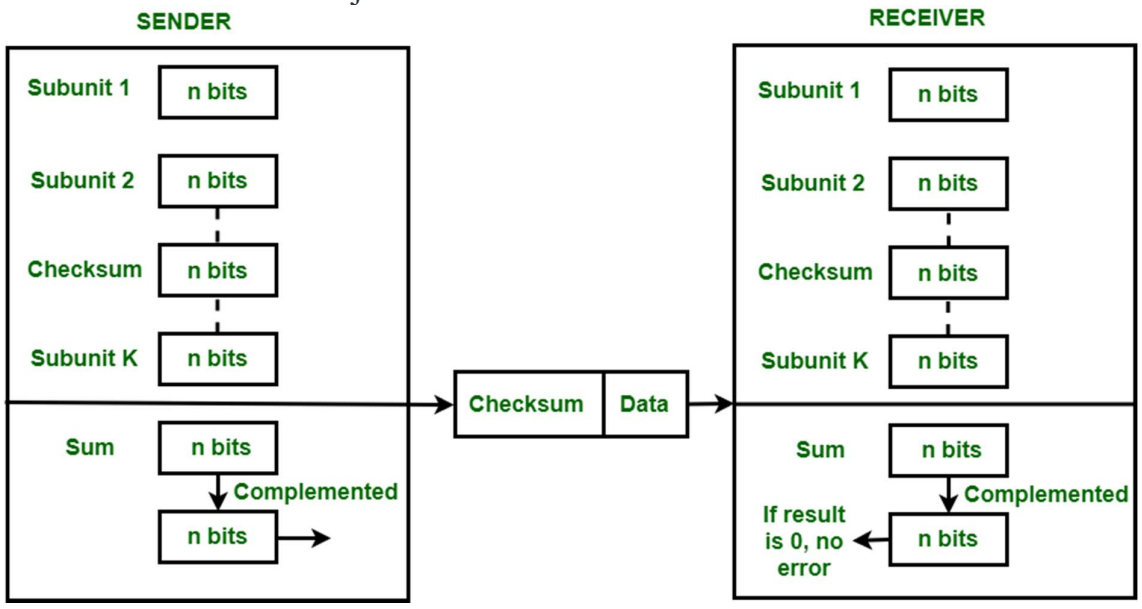
**3. Write a Program to implement data link layer framing method checksum.**

**Description:**

**Checksum** is the error detection method used by upper-layer protocols and is considered to be more reliable than Longitudinal Redundancy Check (LRC), Vertical Redundancy Check (VRC), and Cyclic Redundancy Check (CRC). This method uses a **Checksum Generator** on the sender side and a **Checksum Checker** on the receiver side.

It is a unique number generated from data to verify its integrity. When data is created, a checksum is calculated and sent or saved with it. Later, when accessing the data, the checksum is recalculated. If the two checksums match, the data is likely error free. In this article we will see checksum error detection method in detail.

On the Sender side, the data is divided into equal subunits of n bit length by the checksum generator. This bit is generally of 16-bit length. These subunits are then added together using one's complement method. This sum is of n bits. The resultant bit is then complemented. This complemented sum which is called checksum is appended to the end of the original data unit and is then transmitted to the receiver. The Receiver after receiving data + checksum passes it to checksum checker. Checksum checker divides this data unit into various subunits of equal length and adds all these subunits. These subunits also contain checksum as one of the **subunits**. The resultant bit is then complemented. If the complemented result is zero, it means the data is error-free. If the result is non-zero it means the data contains an error and Receiver rejects it.



**PROGRAM:**

```
s1 = input("Enter the string of 0's and 1's as subunit1: ")
s2 = input("Enter the string of 0's and 1's as subunit2: ")
s1 = s1[::-1]
s2 = s2[::-1]

res = ""
c = '0'

for i,j in zip(s1, s2):
if i == '0' and j == '0' and c == '0':
res += '0'
c = '0'
elif i == '0' and j == '0' and c == '1':
res += '1'
c = '0'
elif i == '0' and j == '1' and c == '0':
res += '1'
c = '0'
elif i == '0' and j == '1' and c == '1':
res += '0'
c = '1'
elif i == '1' and j == '0' and c == '0':
res += '1'
c = '0'
elif i == '1' and j == '0' and c == '1':
res += '0'
c = '1'
elif i == '1' and j == '1' and c == '0':
res += '0'
c = '1'
elif i == '1' and j == '1' and c == '1':
res += '1'
c = '1'


if c == '1':
ans = ""
for i in res:
if i == '1' and c == '1':
ans += '0'
c = '1'
elif i == '0' and c == '0':
```

```
ans += '0'
c = '0'
else :
ans += '1'
c = '0'
res = ans

final = ""
for i in res:
if i == '1':
final += '0'
else:
final += '1'
print("Checksum of two subunits: ", final[::-1].strip())
```

**OUTPUT:**

```
Enter the string of 0's and 1's as subunit1: 10101001

Enter the string of 0's and 1's as subunit2: 00111001

Checksum of two subunits: 00011101
```

## EXPERIMENT 4

**4. Write a program for Hamming Code generation for error detection and correction.**