Government College of Engineering, Jalgaon

(An Autonomous Institute of Govt. of Maharashtra)
Department of Computer Engineering

Student Name   :_____ **PRN:** _____

Course Teacher   : Mrs Priyanka H. Gadade, Government College of Engg., Jalgaon

Experiment No. ____

Title: Implementation of Internet checksum

Aim: To Implementation of Internet Checksum

THEORY:-

The Internet Checksum is used for detecting corruption in a block of data .It is initialized to zero, and computed as the complement of the ones-complement sum of the data, taken in 16- bit units. A subsequent checksum of the data and checksum together should generate a zero checksum if no errors are detected.
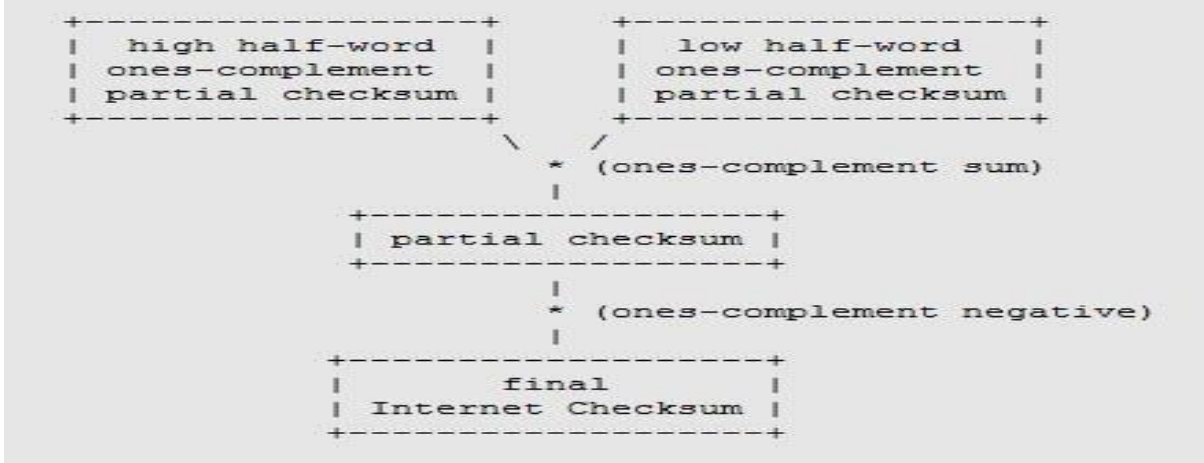
The checksum allows:

- Byte order "independence" reordered output is equivalent to reordered input

- 16-bit word-order independence reordering 16-bit words preserves the output

- Incremental computation

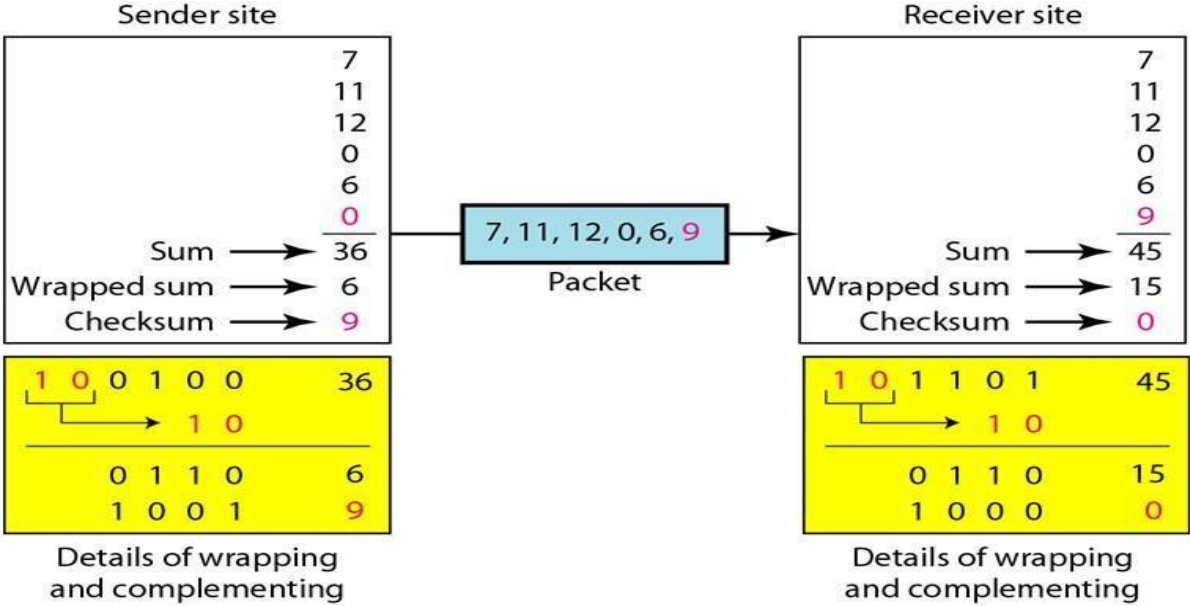- deferred carries

- parallel summation

A result of deferred carries, incremental Computation, and 16-bit word order independence

This note describes an implementation that computes two partialchecksums in parallel, over the odd and even 16-bit half-words of32-bit data. The result is a pair of partial checksums (odd and even), which can be combined, and the result inverted to generate thetrue Internet Checksum.

This technique is related to the long-wordparallel summation used in efficient software implementations.

```
+--------------------------+         +--------------------------+
|      high half-word      |         |      low half-word       |
|     ones-complement      |         |     ones-complement      |
|    partial checksum      |         |    partial checksum      |
+--------------------------+         +--------------------------+
                        \         /
                         *  (ones-complement sum)
                         |
            +---------------------------+
            |     partial checksum      |
            +---------------------------+
                         |
                         *  (ones-complement negative)
                         |
            +---------------------------+
            |           final           |
            |     Internet Checksum     |
            +---------------------------+
```

Example of Checksum

Sender site   Receiver site

	7			7
	11			11
	12			12
	0			0
	6			6
	0			9
Sum →	36	Packet: 7, 11, 12, 0, 6, 9	Sum →	45
Wrapped sum →	6		Wrapped sum →	15
Checksum →	9		Checksum →	0

Details of wrapping and complementing (Sender):

```
1 0 0 1 0 0      36
      → 1 0
_____
0 1 1 0           6
1 0 0 1           9
```

Details of wrapping and complementing (Receiver):

```
1 0 1 1 0 1      45
      → 1 0
_____
0 1 1 0          15
1 0 0 0           0
```

Procedure to calculate the Internet Checksum Sender site:

1. The message is divided into 16-bit words.

2. The value of the checksum word is set to 0.

3. All words including the checksum are added using one's complement addition.

4. The sum is complemented and becomes the checksum.

5. The checksum is sent with the data.

Receiver site:

1. The message (including checksum) is divided into 16-bit words.

2. All words are added using one's complement addition.

3. The sum is complemented and becomes the new checksum.

4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

Example of Internet Checksum
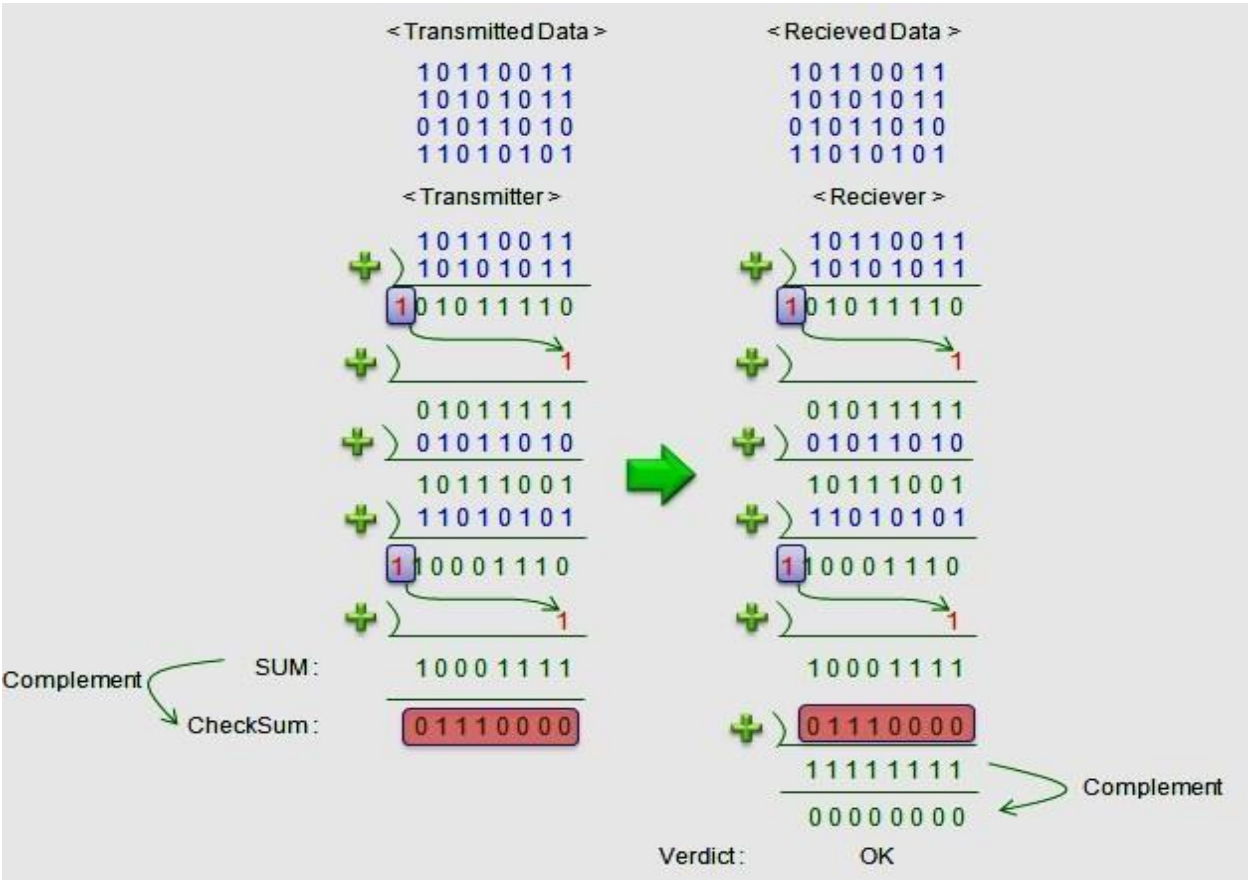
Program for checksum

```c
#include<stdio.h>
int add(int,int);
int com(int);
int main()
{       int i,j,dl,dil;
        int data1[10],data2[10],newdata[10],comp[10],checksum[10];
        printf("\n Enter the data length = ");
        scanf("%d",&dl);
        printf("\n Enter the data1 : \n");
        for(i=0;i<dl;i++)
                scanf("%d",&data1[i]);
        printf("\n Enter the data2 : \n");
        for(i=0;i<dl;i++)
                scanf("%d",&data2[i]);
        for(i=dl-1;i>=0;i--)
        {       newdata[i]=add(data1[i],data2[i]);
        }
        printf("\n Data 1      : ");
        for(i=0;i<dl;i++)
                printf("%d",data1[i]);

        printf("\n Data 2      : ");
        for(i=0;i<dl;i++)
                printf("%d",data2[i]);

        printf("\n\n The new data is : ");
        for(i=0;i<dl;i++)
                printf("%d",newdata[i]);


        printf("\n Checksum : ");
        for(i=0;i<dl;i++)
        {       checksum[i]=com(newdata[i]);
                printf("%d",checksum[i]);
        }
        printf("\n\n Receiver Side : \n");

        printf("\n Data : ");
        for(i=0;i<dl;i++)
                printf("%d",data1[i]);

        printf(" ");
        for(i=0;i<dl;i++)
                printf("%d",data2[i]);
```

```
        printf(" ");
        for(i=0;i<dl;i++)
                printf("%d",checksum[i]);

        printf("\n Addition : ");
        for(i=dl-1;i>=0;i--)
        {       newdata[i]=add(newdata[i],checksum[i]);
        }
        for(i=0;i<dl;i++)
        {       printf("%d",newdata[i]);
        }

        printf("\n  Compliment : ");
        for(i=0;i<dl;i++)
                comp[i]=com(newdata[i]); printf("%d",comp[i]);
        return(0);
}

int add(int x, int y)
{       static int carry=0;
        if(x =  =1 && y = =1 && carry = =0)
        {       carry=1;
                return 0;
        }
        else if(x==1 && y==1 && carry==1)
        {
                carry=1;
                return 1;
        }
        else if(x==1 && y==0 && carry==0)
        {       carry=0;
                return 1;
        }
        else if(x==1 && y==0 && carry==1)
        {       carry=1;
                return 0;
        }
        else if(x==0 && y==1 && carry==0)
        {       carry=0;
                return 1;
        }

        else if(x==0 && y==1 && carry==1)
        {       carry=1;
```

```
                return 0;
        }
        else if(x==0 && y==0 && carry==0)
        {       carry=0;
                return 0;
        }
        else
        {       carry=0;
                return 1;
        }
}

int com(int a)
{       if(a==0)
                return 1;
        else
                return 0;
}
```

OUTPUT:

harish@harish-Inspiron-3537:~/harish$ gcc checksum.c
harish@harish-Inspiron-3537:~/harish$ ./a.out

Enter the data length = 8

Enter the data1 :
1 0 1 1 0 0 1 1

Enter the data2 :
1 0 1 0 1 0 1 1

Data 1   : 10110011
Data 2   : 10101011

The new data is : 01011110
Checksum : 10100001

Receiver Side :

Data : 10110011 10101011 10100001
Addition : 00000000
Compliment : 1
harish@harish-Inspiron-3537:~/harish$

POST LAB TASK:-

1. What is checksum?

_____

_____

_____

2. Apply the checksum error detection technique at sending end to the following bit stream? 11101011010010

_____

_____

_____

Mrs. Priyanka H. Gadade
Course Teacher