COL788 REPORT
ASSIGNMENT1

2018CS50408
K LAXMAN

A Technique used for finding unintentional modifications or mistakes in the Communication channel is called cyclic redundancy check.Generator polynomial Which are available on both the sender and receiver sides ,are used by CRC.

- I have used C++ code to compute the cyclic redundancy check (CRC) and given the input first with 8 ASCII Characters .

Below are steps to run my code :

- **RUN:**
  g++ -o assign1 assign1.cpp
  ./assign

- Correspondingly in the **Raspberry Pi os terminal** ,the I have compiled using the command

  arm-linux-gnueabihf-g++ -o 2018CS50408_arm 2018CS50408_assign1.cpp

  ./2018CS50408_arm

- **Terminal Interface looks like:**

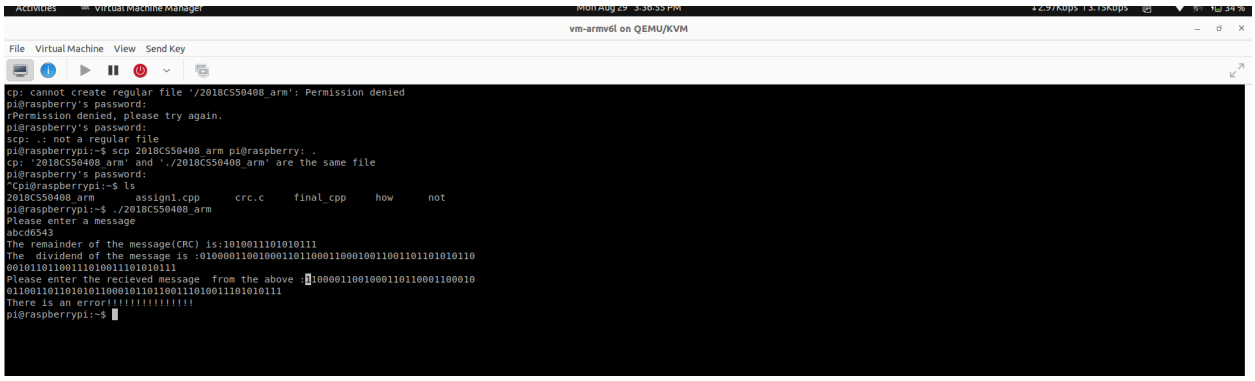
  -Firstly it asks the user to enter the message which has to be sent and it has to be of size 8 ASCII Characters(Accepting both the character and numbers)
  -Once the message is entered it shows the remainder(CSC) of the message which is decoded

- It shows the dividend(message + Remainder(CRC)) and the user is asked to enter the received message from the above .
  Below I have attached the required screenshots of terminals(both ubuntu & Raspberry)

TEST CASES FOR DETECTING ERRORS :

CRC detects all the bursty fault errors which are not longer than n bits like single bit error and odd number of bits.

SINGLE BIT ERROR:
Each and every time a single bit is toggled/changed ,error is thrown from the receiver side as it is using modulo 2 binary division

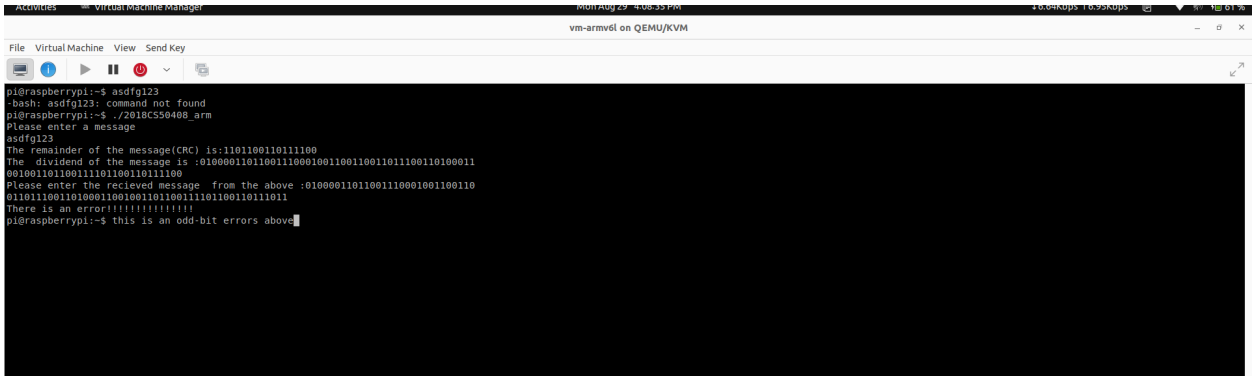


From the above we can see that if the dividend part result is toggled with an change of 1 (original is 0).then the receiver is throwing error

BURST(ODD BIT) ERROR:

CRC can detect burst errors of fewer than r+1 bits.where r is the degree of the polynomial.

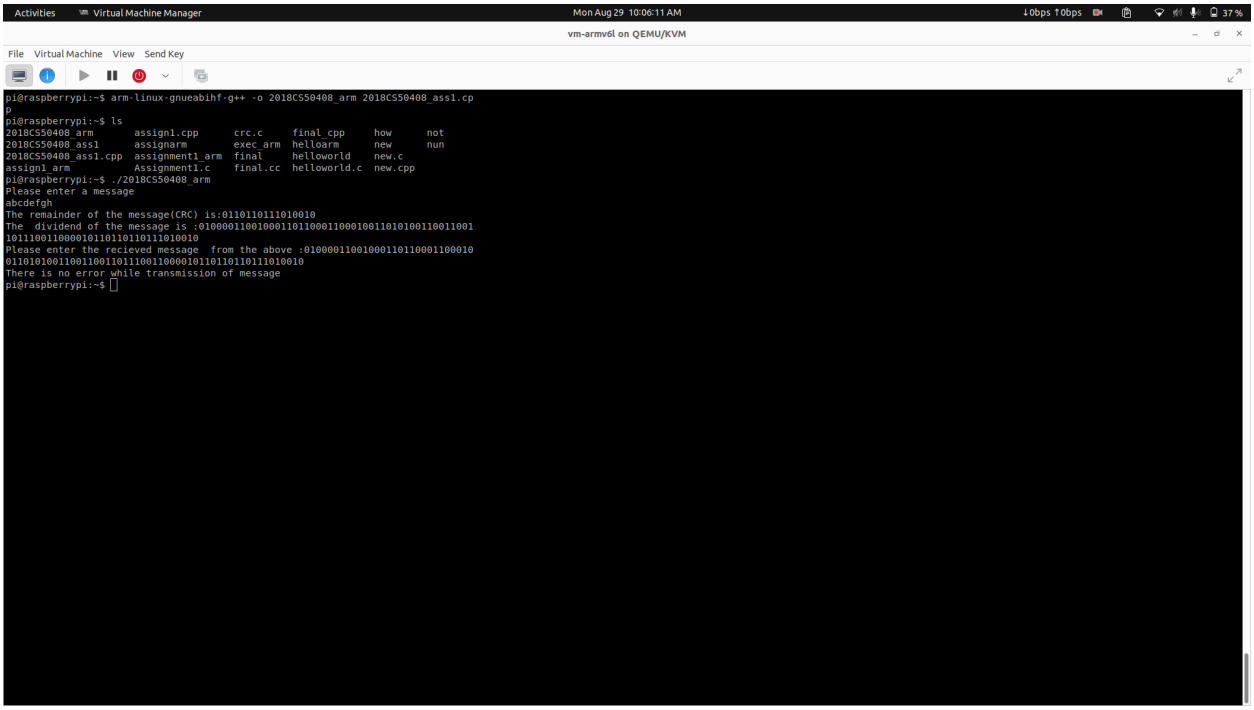
These errors may be a due to physical damage such as scratch on a disc or a stroke of lightning
In case of wireless channels.

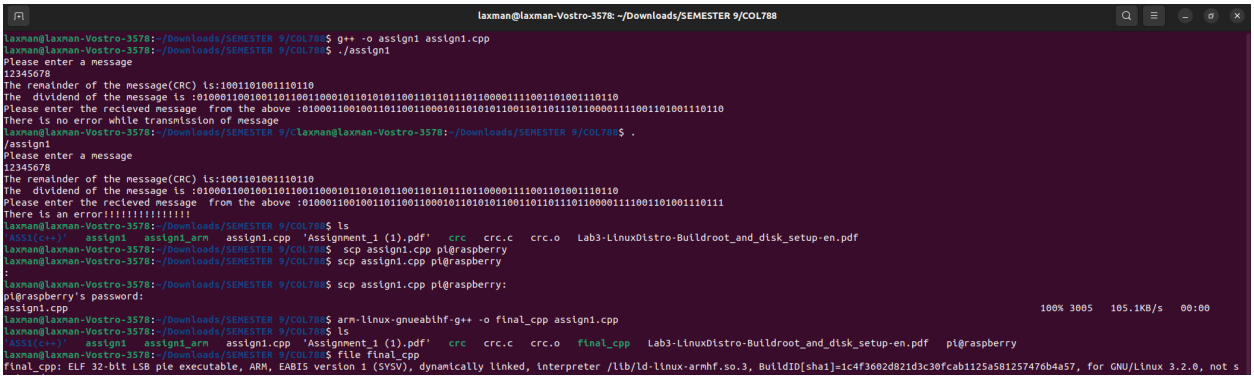


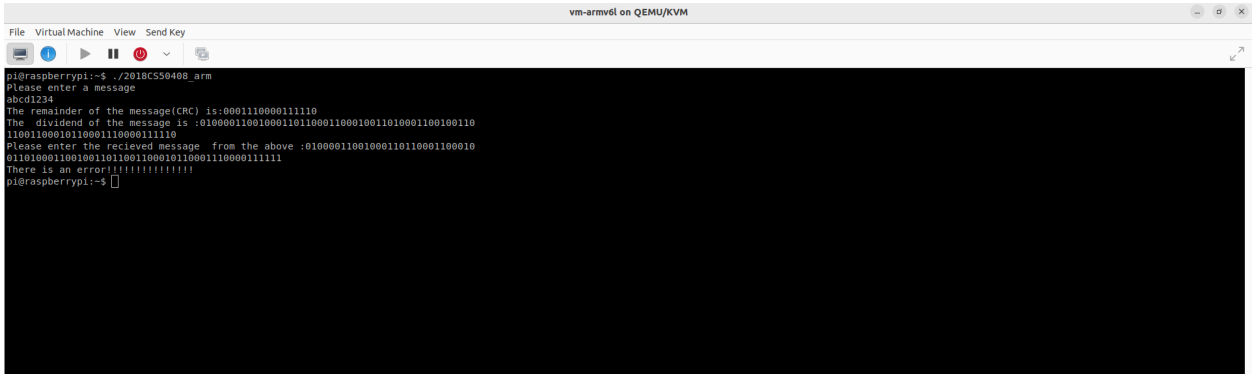
In both of these errors the receiver will give a non zero remainder,which shows that there is an presence of error,i.e. The message has not been communicated correctly or got corrupted.

TEST CASES FOR NOT DETECTING ERRORS :

- A P(x) of degree N will be able to identify bit mistakes and bursts up to N bits long. The probability that a completely incorrect bit pattern will still result in a zero residual is one in 2^N for random bit defects longer than N-bits.
- any bit error term E(x) which is an exact multiple of P(x) will not be detected.here E(x) is an error bit and P(x) is polynomial degree.
- **OUTPUT:**


Activities
Virtual Machine Manager
Mon Aug 29 10:06:11 AM
0bps 0bps
37 %
vm-armv6l on QEMU/KVM
File Virtual Machine View Send Key
pi@raspberrypi:~$ arm-linux-gnueabihf-g++ -o 2018CS50408_arm 2018CS50408_ass1.cp
p
pi@raspberrypi:~$ ls
2018CS50408_arm assign1.cpp crc.c final_cpp how not
2018CS50408_ass1 assignarm exec_arm helloarm new nun
2018CS50408_ass1.cpp assignment1_arm final helloworld new.c
assign1_arm Assignment1.c final.cc helloworld.c new.cpp
pi@raspberrypi:~$ ./2018CS50408_arm
Please enter a message
abcdefgh
The remainder of the message(CRC) is:0110110111010010
The dividend of the message is :0100001100100011011000110001001101010011001 1
1011100110000101101101101110100 10
Please enter the recieved message from the above :0100001100100011011000110001 0
0110101001100110011011100110000101101101101110100 10
There is no error while transmission of message
pi@raspberrypi:~$


laxman@laxman-Vostro-3578: ~/Downloads/SEMESTER 9/COL788
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ g++ -o assign1 assign1.cpp
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ ./assign1
Please enter a message
12345678
The remainder of the message(CRC) is:1001101001110110
The dividend of the message is :0100011001001101100100101101010110011011011100001111001101001110110
Please enter the recieved message from the above :0100011001001101100100101101010110011011011100001111001101001110110
There is no error while transmission of message
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/Claxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$.
/assign1
Please enter a message
12345678
The remainder of the message(CRC) is:1001101001110110
The dividend of the message is :0100011001001101100100101101010110011011011100001111001101001110110
Please enter the recieved message from the above :0100011001001101100100101101010110011011011100001111001101001110111
There is an error!!!!!!!!!!!!!!
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ ls
'ASS1(c++)' assign1 assign1_arm assign1.cpp 'Assignment_1 (1).pdf' crc crc.c crc.o Lab3-LinuxDistro-Buildroot_and_disk_setup-en.pdf
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ scp assign1.cpp pi@raspberry
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ scp assign1.cpp pi@raspberry:
pi@raspberry's password:
assign1.cpp 100% 3005 105.1KB/s 00:00
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ arm-linux-gnueabihf-g++ -o final_cpp assign1.cpp
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ ls
'ASS1(c++)' assign1 assign1_arm assign1.cpp 'Assignment_1 (1).pdf' crc crc.c crc.o final_cpp Lab3-LinuxDistro-Buildroot_and_disk_setup-en.pdf pi@raspberry
laxman@laxman-Vostro-3578:~/Downloads/SEMESTER 9/COL788$ file final_cpp
final_cpp: ELF 32-bit LSB pie executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, BuildID[sha1]=1c4f3602d821d3c30fcab1125a581257476b4a57, for GNU/Linux 3.2.0, not s


vm-armv6l on QEMU/KVM
File Virtual Machine View Send Key
pi@raspberrypi:~$ ./2018CS50408_arm
Please enter a message
abcd1234
The remainder of the message(CRC) is:0001110000111110
The dividend of the message is :0100001100100011011000110001001101010001100100110
1100110001011000111000011111 0
Please enter the recieved message from the above :0100001100100011011000110001 0
0110100011001001101100110001011000111000011111
There is an error!!!!!!!!!!!!!!
pi@raspberrypi:~$