PROJECT REPORT ON ERROR DETECTION TECHNIQUES IMPLEMENTED IN KEIL

Submitted by :

Shubham Babel (MT2018522)
Siva Mounika (MT2018523)
Swagatika Mahapatra (MT2018525)

Error Detection:

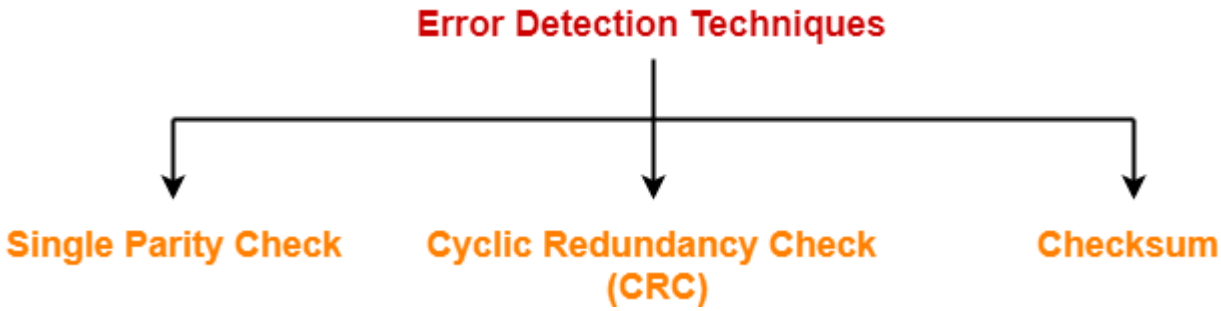
When sender transmits data to the receiver, the data might get scrambled by noise or data might get corrupted during the transmission.

Error detection is a technique that is used to check if any error occurred in the data during the transmission.

Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if an error occurred during transmission of the message.

Error Detection Methods

Some popular error detection methods are



1. Checksum

A checksum is an error-detection method in a the transmitter computes a numerical value according to the number of set or unset bits in a message and sends it along with each message frame. At the receiver end, the same checksum function (formula) is applied to the message frame to retrieve the numerical value. If the received checksum value matches the sent value, the transmission is considered to be successful and error-free.

Error detection using checksum method involves the following steps-

Step-01:

At sender side,

- If m bit checksum is used, the data unit to be transmitted is divided into segments of m bits.
- All the m bit segments are added.
- The result of the sum is then complemented using 1's complement arithmetic.
- The value so obtained is called as **checksum**.

Step-02:

- The data along with the checksum value is transmitted to the receiver.

Step-03:

At receiver side,

- If m bit checksum is being used, the received data unit is divided into segments of m bits.
- All the m bit segments are added along with the checksum value.
- The value so obtained is complemented and the result is checked.

Then, following two cases are possible

Case-01: Result = 0

If the result is zero,

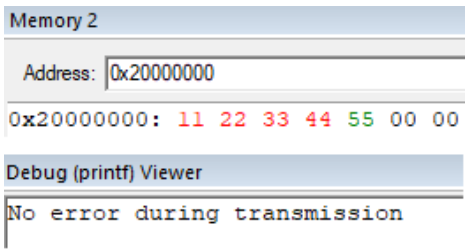
- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

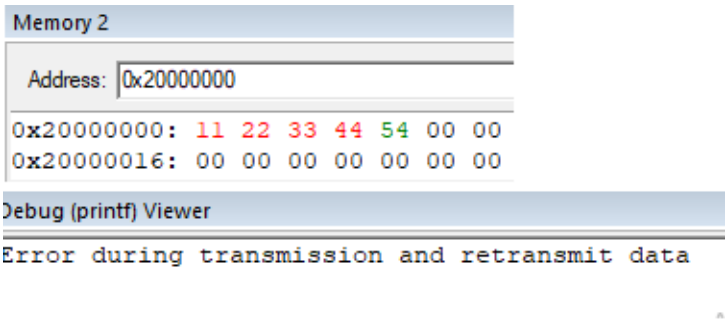
Case-02: Result ≠ 0

If the result is non-zero,

- Receiver assumes that error occurred in the data during the transmission.
- Receiver discards the data and asks the sender for retransmission.

Results:





2. Single Parity Check

In this technique,

- One extra bit called as **parity bit** is sent along with the original data bits.
- Parity bit helps to check if any error occurred in the data during the transmission.

Steps Involved

Error detection using single parity check involves the following steps

Step-01:

At sender side,

- Total number of 1's in the data unit to be transmitted is counted.
- The total number of 1's in the data unit is made even in case of even parity.
- The total number of 1's in the data unit is made odd in case of odd parity.
- This is done by adding an extra bit called as **parity bit**.

Step-02:

- The newly formed code word (Original data + parity bit) is transmitted to the receiver.

Step-03:

At receiver side,

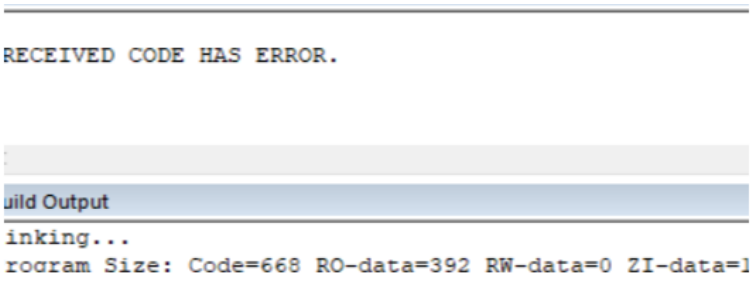
- Receiver receives the transmitted code word.
- The total number of 1's in the received code word is counted.

Then, following cases are possible-

- If total number of 1's is even and even parity is used, then receiver assumes that no error occurred.
- If total number of 1's is even and odd parity is used, then receiver assumes that error occurred.
- If total number of 1's is odd and odd parity is used, then receiver assumes that no error occurred.
- If total number of 1's is odd and even parity is used, then receiver assumes that error occurred.

In printf viewer, if output = 1 then received code is erroneous else the code is correct what is sent
We have implemented even parity.

Results



3. Cyclic Redundancy Check

Cyclic Redundancy Check (CRC) is an error detection method. It is based on binary division.

CRC generator is an algebraic polynomial represented as a bit pattern. Bit pattern is obtained from the CRC generator using the following rule-

Step-01: Calculation Of CRC At Sender Side

At sender side,

- A string of n 0's is appended to the data unit to be transmitted.
- Here, n is one less than the number of bits in CRC generator.
- Binary division is performed of the resultant string with the CRC generator.
- After division, the remainder so obtained is called as **CRC**.

- It may be noted that CRC also consists of n bits.

Step-02: Appending CRC To Data Unit-

At sender side,

- The CRC is obtained after the binary division.
- The string of n 0's appended to the data unit earlier is replaced by the CRC remainder.

Step-03: Transmission To Receiver

- The newly formed code word (Original data + CRC) is transmitted to the receiver.

Step-04: Checking at Receiver Side

At receiver side,

- The transmitted code word is received.
- The received code word is divided with the same CRC generator.
- On division, the remainder so obtained is checked.

The following two cases are possible

Case-01: Remainder = 0

If the remainder is zero,

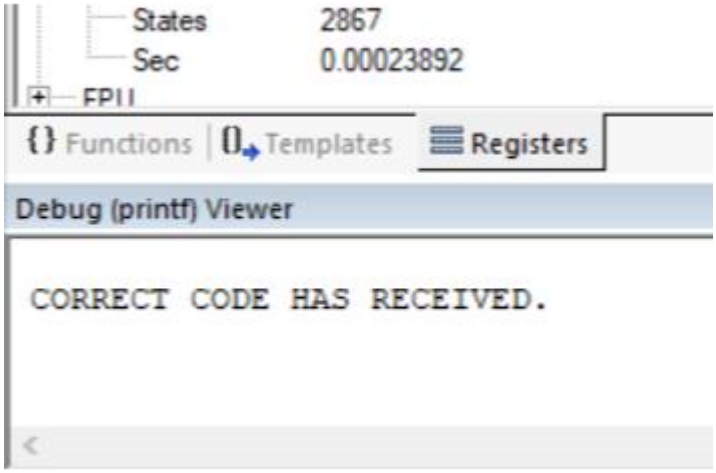
- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Remainder ≠ 0

If the remainder is non-zero,

- Receiver assumes that some error occurred in the data during the transmission.
- Receiver rejects the data and asks the sender for retransmission.

Result


States 2867
Sec 0.00023892
FPU
Functions
Templates
Registers
Debug (printf) Viewer
CORRECT CODE HAS RECEIVED.