CRC Pattern

Hazırlayan:

Mehmet Faruk Gül - 031790044

CRC Pattern Nedir?

- 1961 yılında W. Wesley Peterson tarafından tasarlanmış bir blok kodudur.
- CRC'nin açılımı türkçede Döngüsel Artıklık Kontrolü olarak geçmektedir. CRC'lerin böyle adlandırılmasının sebebi konrol değerlerinin bir fazlalık yani mesajı genişleten bir bilgi olmasından dolayıdır ve algoritması döngüsel kodlara dayanmaktadır.
- Genel anlamda telekomünikasyon ağları ve depolama aygıtları aracılığıyla iletilen verilerdeki kazara değişiklikleri tespit etmek için kullanılmaktadırlar.

CRC Pattern Nasıl Çalışmaktadır?

- Bir CRC hesaplamasının teorisi açıktır. Veriler, CRC algoritması tarafından ikili sayı olarak ele alınır. Bu sayı, polinom adı verilen başka bir ikili sayıya bölünür. Bölümün geri kalanı, iletilen mesaja eklenen CRC sağlama toplamıdır. Alıcı, mesajı (hesaplanan CRC dahil), kullanılan vericiyle aynı polinomla böler. Bu bölmenin sonucu sıfır ise, iletim başarılı olmuştur.

Örnek 1

- Kodlanacak mesajımız: 11010011101100
- Bu, ilk olarak CRC'nin bit uzunluğuna n karşılık gelen sıfırlarla doldurulur . Bu, ortaya çıkan kod sözcüğünün sistematik biçimde olması için yapılır. İşte 3 bitlik bir CRC'yi hesaplamak için ilk hesaplama:

```
11010011101100 000 <--- input right padded by 3 bits
1011               <--- divisor (4 bits) = x³ + x + 1
------------------
01100011101100 000 <--- result
```

Örnek 1 Devam

- Algoritma, her adımda doğrudan bölenin üzerindeki bitlere etki eder. Bu yinelemenin sonucu, polinom böleninin üzerindeki bitlerle bitsel XOR değeridir. Bölenin üstünde olmayan bitler, o adım için doğrudan aşağıya kopyalanır. Bölen daha sonra girişte kalan en yüksek 1 bit ile hizalanacak şekilde sağa kaydırılır ve bölen giriş satırının sağ ucuna ulaşıncaya kadar işlem tekrarlanır. İşte tüm hesaplama yan sayfada gösterilmektedir:

Örnek 1 Devam

```
11010011101100 000 <--- input right padded by 3 bits
1011               <--- divisor
01100011101100 000 <--- result (note the first four bits are the XOR with the div
unchanged)
 1011              <--- divisor ...
00111011101100 000
  1011
00010111101100 000
   1011
00000001101100 000 <--- note that the divisor moves over to align with the next 1
step was zero)
       1011            (in other words, it doesn't necessarily move one bit per
00000000110100 000
        1011
00000000011000 000
         1011
00000000001110 000
          1011
00000000000101 000
           101 1
-----------------
00000000000000 100 <--- remainder (3 bits).  Division algorithm stops here as div
```

- En soldaki bölen bit dokunduğu her giriş bitini sıfırladığından, bu işlem sona erdiğinde giriş satırındaki sıfır olmayan tek bit, satırın sağ ucundaki n bittir. Bu n bit, bölme adımının geri kalanıdır ve aynı zamanda CRC fonksiyonunun değeri olacaktır (seçilen CRC spesifikasyonu bir miktar sonradan işleme gerektirmedikçe).

Örnek 1 Son

```
11010011101100 100 <--- input with check value
1011               <--- divisor
01100011101100 100 <--- result
 1011              <--- divisor ...
00111011101100 100

......

00000000001110 100
          1011
00000000000101 100
           101 1
----------------
00000000000000 000 <--- remainder
```

- Alınan bir mesajın geçerliliği, yukarıdaki hesaplama tekrar yapılarak, bu sefer sıfır yerine kontrol değeri eklenerek kolaylıkla doğrulanabilir. Saptanabilir hata yoksa kalan sıfıra eşit olmalıdır.

Örnek 2

- İkili biçim: 1001100, 110111'e bölme işlemini yapacağız.
- Bölümün polinom hali:

x^5 + x^4 + x^2 + x + 1

```
İkili biçim (sıfır eklendi): 100110000000 bölü 110111
Sonuç 1111010
Kalan 00110
Çalışma:

     1111010
------------
100110000000
110111
------
 10001000000
 110111
 ------
  1010100000
  110111
  ------
   111010000
   110111
   ------
    01101000
    000000
    ------
     1101000
     110111
     ------
      000110
      000000
      ------
       00110

İletilen değer: 100110000110
```

Örnek 2 Devam

Şimdi değerimizi aldık ve check işlemimizi yapıyoruz yanda. Yaptığımız işlem sonucunda değerimiz 0 çıktığından dolayı bir hata olmadığını anlıyoruz.

```
İkili Form:     100110000110 / 110111

İkili biçim (sıfır eklendi):   10011000011000000 / 110111
Sonuç:  111101000000
Kalan:  00000

Çalışma:
-----------------
10011000011000000
110111
------
 1000100011000000
 110111
 ------
  101010011000000
  110111
  ------
   11101011000000
   110111
   ------
    01101111000000
    000000
    ------
     1101111000000
     110111
     ------
      000000000000
      000000
      ------
       0000000000
       000000
       ------
        000000000
        000000
        ------
         00000000
         000000
         ------
          0000000
          000000
          ------
           000000
           000000
           ------
            00000
```

Teşekkür ederim

Hazırlayan:

Mehmet Faruk Gül - 031790044