Описание и расчёт CRC16

CRC (*Cyclic Redundancy Code - циклический избыточный код*) - алгоритм расчёта контрольной суммы для передаваемого сообщения, основанный на полиномиальной арифметике.

Основная идея алгоритма CRC состоит в представлении сообщения в виде огромного двоичного числа, делении его на другое фиксированное двоичное число и использовании остатка от этого деления в качестве контрольной суммы. Получив сообщение, приёмник должен выполнить аналогичное действие и сравнить полученный результат с принятой контрольной суммой. Сообщение считается достоверным, выполняется это равенство.

Алгоритм CRC базируется на полиномиальной арифметике, а это означает, что сообщение, делитель и остаток могут быть представлены в виде полиномов с двоичными коэффициентами или в виде строки битов, каждый из которых является коэффициентом полинома.

Кроме того, нужно отметить, что в CRC алгоритме используется полиномиальная арифметика по модулю 2. Это означает, что действия, выполняемые во время вычисления CRC, являются арифметическими операциями без учёта переноса. То есть сложение и вычитание выполняются побитово без учёта переноса, благодаря чему эти две операции дают эквивалентный результат. Операции сложения и вычитания в этом случае идентичны операции XOR (исключающее ИЛИ).

Деление выполняется по аналогии с обычным арифметическим делением столбиком с тем отличием, что вместо вычитания делителя из делимого используется операция XOR. В программной реализации этого алгоритма вместо делителя сдвигается делимое. Сдвиг осуществляется влево по одному биту. При этом выполняется проверка выдвигаемого бита: если он равен единице, выполняется операция XOR делителя (полинома) со старшими разрядами делимого (сообщения).

Чтобы выполнить вычисление CRC, необходимо выбрать делитель - полином. Важной характеристикой, определяющей дальнейшие расчёты, является степень полинома или его ширина W (от английского Width - ширина). Обычно выбирается степень 16 или 32, так как они являются кратными разрядности регистров современных процессоров, что значительно упрощает реализацию алгоритмов CRC.

Степень полинома - действительная позиция старшего бита. Позиции битов отсчитываются, начиная с нулевой.

Выбрав полином, нужно в конец сообщения добавить W нулевых битов.

В данной статье для расчета CRC будет достаточно использовать полином степени W = 16, который обеспечивает вероятность не обнаружения ошибки $\frac{1}{2^{16}} \approx 15{,}3 \cdot 10^{-6}$. При том, что ошибки возникают тоже с достаточно малой вероятностью, доля искажённых данных среди всех принятых на хранение компьютером, будет очень мала и не окажет значительного влияния на дальнейшую работу с сохранёнными в компьютере данными.

Существует набор стандартных полиномов степени W = 16. В данном проекте будем использовать стандартный полином CRC-16, имеющий значение "8005" в шестнадцатеричном представлении или 1 1000 0000 0000 0101 в двоичном представлении (как правило, самая старшая единица не учитывается при расчётах).

При расчётах CRC используется абстрактный регистр CRC с разрядностью, равной степени полинома W, который хранит текущее значение вычисляемой контрольной суммы. Кроме степени полинома нужно выбрать начальное значение регистра CRC и значение, которое комбинируется по XOR с окончательным содержимым регистра. Лучшим выбором для инициализации регистра является значение FFFFh (или 1111 1111 1111 1111 в двоичном формате), которое обеспечить возможность обнаружить нулевые байты. Соответственно, окончательное значение регистра будет комбинироваться по XOR также с FFFFh.

Простой алгоритм расчёта CRC выполняется следующим образом:

1. В регистр CRC заносится начальное значение FFFFh.

2. В конец сообщения добавляется W нулевых битов

3. Содержимое регистра сдвигается влево на 1 бит, и в последнюю (нулевую) позицию заносится очередной, ещё не обработанный бит данных.

4. Если из регистра был выдвинут бит со значением "1", то содержимое регистра комбинируется по XOR с полиномом. Если значение бита равно "0", XOR не выполняется.

5. Шаги 3 и 4 выполняются, пока не будут обработаны все данные.

6. Окончательное содержимое регистра комбинируется по XOR со значением FFFFh.

Примечание: в программной реализации сначала выполняется проверка старшего бита, и только потом - сдвиг содержимого регистра.

Для ускорения расчёта CRC используется табличный алгоритм. Его суть состоит в следующем: при выполнении операции XOR содержимого регистра с постоянной величиной при различных её сдвигах всегда будет существовать некоторое значение, которое при применении операции XOR с исходным содержимым регистра даст тот же самый результат. А значит, можно составить таблицу таких величин, где индексом является исходное содержимое регистра. Эта таблица позволяет значительно ускорить расчёт CRC заменой восьми операций сдвига одной операцией поиска по таблице.

Всего значений в таблице 256. Поэтому при ее расчёте выполняется цикл по 256 значениям (от 0 до 255):

1.Очередной индекс помещается в регистр. Так как индекс представляет собой один байт, а величина полинома и регистра - два байта, индекс помещается в старший байт регистра, а младший байт заполнен нулями.

2. Содержимое регистра восемь раз сдвигается влево по одному биту. Каждый раз проверяется выдвинутый бит. Если из регистра был выдвинут бит со значением "1", то содержимое регистра комбинируется по XOR с полиномом. Если значение бита равно "0", XOR не выполняется.

3. Полученная двухбайтовая величина заносится в таблицу по индексу.

После того как таблица рассчитана, можно использовать табличный алгоритм CRC:

1. Сдвиг регистра на 1 байт влево с чтением нового байта сообщения.

2. XOR старшего байта, только что выдвинутого из регистра с новым байтом сообщения, что даёт индекс в таблице [0..255].

3. XOR табличного значения с содержимым регистра.

4. Если ещё есть байты данных, перейти к шагу 1.

В данной статье используется описанный выше табличный алгоритм.

Код для расчёта таблицы:

```
Word MakeCRC16Table(void)
{
 Word r;
  for(int i=0; i<256; i++)
   {
    r = ((Word)i)<<8;
     for(byte j=0; j<8; j++)
       {
        if(r&(1<<15)) r=(r<<1)^0x8005;
        else r=r<<1;
       }
     crctable[i]=r;
   }
}
```

Код для расчёта CRC:

```
Word GetCRC16(byte *buf, Word len)
{
 Word crc;
 crc = 0xFFFF;
 while(len--)
  {
   crc = crctable[((crc>>8)^*buf++)&0xFF] ^ (crc<<8);
  }
 crc ^= 0xFFFF;
 return crc;
}
```

