

Контрольный пример

Вычислим значение алгоритма CRC8-MAXIM для строки «123456789».

Характеристики данного алгоритма:

1) Производящий полином: $x^8 + x^5 + x^4 + 1$

2) Стартовые данные, то есть значения регистров на момент начала вычислений:

0xFF

3) Флаг (*RefIn*), указывающий на начало и направление вычислений: true

4) Флаг (*RefOut*), определяющий, инвертируется ли порядок битов регистра при входе на элемент XOR: true

5) Число (*XorOut*), с которым складывается по модулю 2 полученный результат:

0x00

6) Значение CRC (*check*) для строки «123456789»: 0xA1

Будем выполнять алгоритм поочередно для каждого ASCII-кода символа из строки «123456789».

1) Выполним алгоритм для символа “1”, его ASCII-код: 49 или 0011 0001 в двоичном виде.

Начально значение в регистре для алгоритма CRC8/MAXIM: 0x00.

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и инициализирующего значения, предварительно сделав реверс бит для входных данных, так как флаг *RefIn* равен true. Получится 1000 1100.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	8C	Выполняем xor
2	29	

3	52	
4	A4	Выполняем хог
5	79	
6	F2	Выполняем хог
7	D5	Выполняем хог
8	9B	Выполняем хог

Значение в регистре после последнего преобразования: 0x07

2) Выполним алгоритм для символа “2”, его ASCII-код: 50 или 0011 0010 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0x07.

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг RefIn равен true. Получится 0100 1100.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	4B	
2	96	Выполняем хог
3	1D	
4	3A	
5	74	
6	E8	Выполняем хог
7	E1	Выполняем хог
8	F3	Выполняем хог

Значение в регистре после последнего преобразования: 0xD7

3) Выполним алгоритм для символа “3”, его ASCII-код: 51 или 0011 0011 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0xD7.

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг RefIn равен true. Получится 0001 1100.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	1B	
2	36	
3	6C	
4	D8	Выполняем xor
5	81	Выполняем xor
6	33	
7	66	
8	CC	Выполняем xor

Значение в регистре после последнего преобразования: 0xA9

4) Выполним алгоритм для символа “4”, его ASCII-код: 52 или 0011 0100 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0xA9

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг RefIn равен true. Получится 1000 0101.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	85	Выполняем хог
2	3B	
3	76	
4	EC	Выполняем хог
5	E9	Выполняем хог
6	E3	Выполняем хог
7	F7	Выполняем хог
8	DF	Выполняем хог

Значение в регистре после последнего преобразования: 0x8F

5) Выполним алгоритм для символа “5”, его ASCII-код: 53 или 0011 0101 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0x8F

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг `RefIn` равен `true`. Получится 0010 0011.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	23	
2	46	

3	8C	Выполняем xor
4	29	
5	52	
6	A4	Выполняем xor
7	79	
8	F2	Выполняем xor

Значение в регистре после последнего преобразования: 0xD5

б) Выполним алгоритм для символа “6”, его ASCII-код: 54 или 0011 0110 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0xD5

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг RefIn равен true. Получится 1100 1001.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	B9	Выполняем xor
2	43	
3	86	Выполняем xor
4	3D	
5	7A	
6	F4	Выполняем xor
7	D9	Выполняем xor
8	83	Выполняем xor

Значение в регистре после последнего преобразования: 0x37

7) Выполним алгоритм для символа “7”, его ASCII-код: 55 или 0011 0111 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0x37

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг RefIn равен true. Получится 1101 1011.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	DB	Выполняем хог
2	87	Выполняем хог
3	3F	
4	7E	
5	FC	Выполняем хог
6	C9	Выполняем хог
7	A3	Выполняем хог
8	77	

Значение в регистре после последнего преобразования: 0xEE

8) Выполним алгоритм для символа “8”, его ASCII-код: 56 или 0011 1000 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0xEE

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг RefIn равен true. Получится 1111 0010.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	F2	Выполняем xor
2	D5	Выполняем xor
3	9B	Выполняем xor
4	7	
5	0E	
6	1C	
7	38	
8	70	

Значение в регистре после последнего преобразования: 0xE0

9) Выполним алгоритм для символа “9”, его ASCII-код: 57 или 0011 1001 в двоичном виде.

Значение в регистре после предыдущих преобразований: 0xE0

Сначала выполним побитовое сложение по модулю 2 ASCII-кода и значения в регистре, предварительно сделав реверс бит для входных данных, так как флаг `RefIn` равен `true`. Получится 0111 1100.

Далее мы 8 раз выполняем побитовый сдвиг в регистре, и если первый бит равен единице, то дополнительно складываем по модулю 2 значение в регистре с полиномом.

№	Регистр (в шестнадцатеричном виде)	Примечание
1	7C	
2	F8	Выполняем xor

3	C1	Выполняем xor
4	B3	Выполняем xor
5	57	
6	AE	Выполняем xor
7	6D	
8	DA	Выполняем xor

Значение в регистре после последнего преобразования: 0x85

В конце выполнения алгоритма сделаем реверс бит в регистре, так как флаг RefOut равен true. И сложим по модулю 2 значение в регистре с XorOut, равному 0x00. Итоговое значение — 0xA1. Что соответствует значению Check.