

ПРОЕКТИРОВАНИЕ МОДУЛЯ АППАРАТНОГО РАСЧЁТА CRC ДЛЯ ИСПОЛЬЗОВАНИЯ В ВЕДОМЫХ УСТРОЙСТВАХ С СИСТЕМНОЙ ШИНОЙ

Курс «Проектирование СнК с программируемой архитектурой»

Авторы:
Любавин Кирилл Дмитриевич
Кузьмин Павел Андреевич

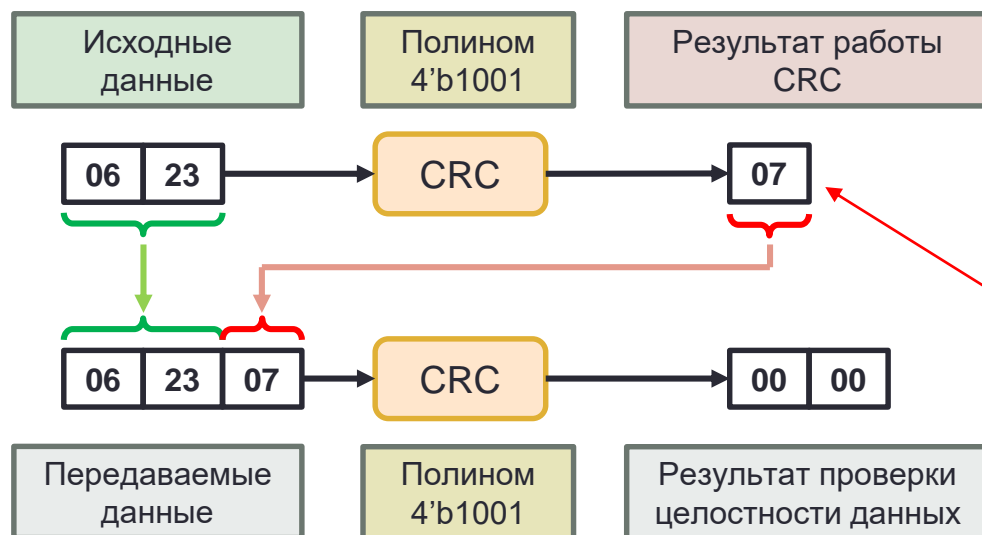
Введение в CRC

Основная идея алгоритма CRC состоит в представлении сообщения виде двоичного числа, делении его на другое фиксированное двоичное число и использовании остатка этого деления в качестве контрольной суммы. Получив сообщение, приёмник может выполнить аналогичное действие и сравнить полученный остаток с "контрольной суммой" (переданным остатком).

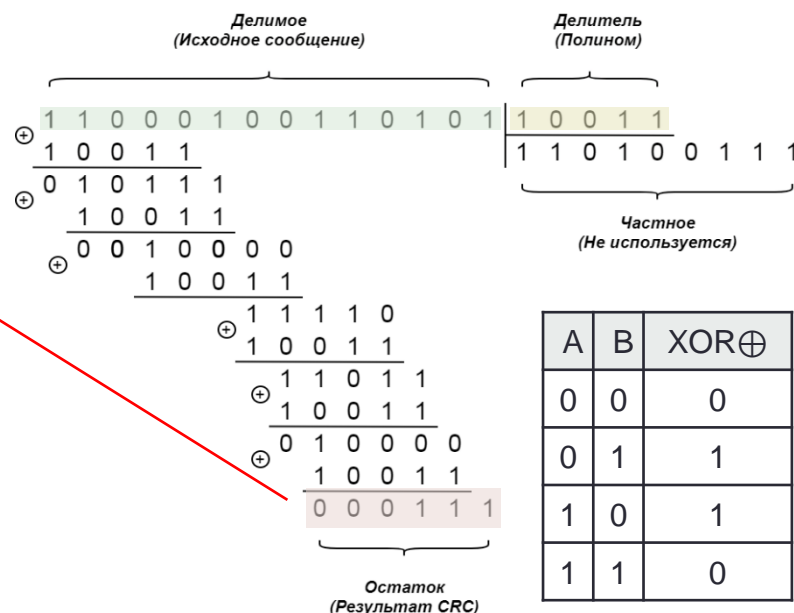
Пусть сообщение (**Делимое**) состоит из 2 байт (**0x6, 0x23**) = **16'b0000_0110_0010_0011**

А ширина регистра контрольной суммы составляет 1 байт, а в качестве полинома (**Делителя**) будет использоваться **4'b1001**

Пример работы CRC



Пример расчёта контрольной суммы



Введение в CRC

Делитель, делимое (сообщения), частное и остаток можно представить в виде полиномов с двоичными коэффициентами или в виде строки бит, каждый из которых является коэффициентом полинома.

$$9_{10} = 4'b1001 \Rightarrow 1 * x^3 + 0 * x^2 + 0 * x^1 + 1 * x^0 \Rightarrow x^3 + 1$$

CRC-арифметика, которая реализует деление исходного пакета данных на требуемый полином соответствует операции «сложение по модулю 2», которое также называется «исключающее или» (XOR).

Размером полинома считается степень самого старшего разряда с прибавлением 1 и имеет обозначение W. В приведённом выше примере размер полинома W равен 4.



A	B	XOR \oplus
0	0	0
0	1	1
1	0	1
1	1	0

Реализация алгоритма вычисления CRC

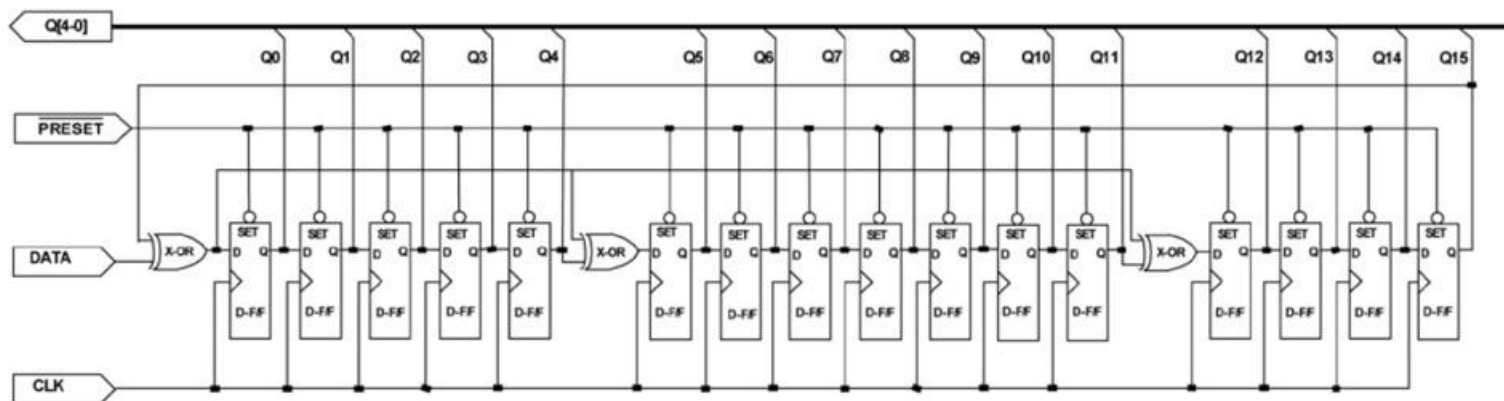
Методика вычисления CRC аппаратным способом сводится к следующему порядку действий

1. Создаётся сдвиговый регистр общей размерностью W ;
2. Регистр инициализируется начальным значением (для каждого типа CRC начальное значение индивидуальное);
3. Исходные данные дополняются логическими нулями в количестве W ;
4. В регистр исходные данные «задвигаются» по одному биту за такт;
5. В случае, если «задвинутое» значение соответствует логической единице, производится операция XOR между полиномом и содержимым сдвигового регистра;
6. После того, как были «задвинуты» все биты исходного сообщения в сдвиговый регистр, в нём будет содержаться остаток от деления, который и будет вычисленным CRC.

Пример аппаратной схемы вычисления CRC-16-CCITT

Одним из самых широко применяемых типов циклически избыточного кода является CRC-16-CCITT. Он широко применяется для передачи данных в оборудовании для телекоммуникаций и СВЧ RFID меток. Полином CRC-16-CCITT соответствует числу $0x1021$. Сам полином выглядит следующим образом:

$$x^{15} + x^{12} + x^5 + x^0$$



Пример схемы на HDL SystemVerilog

Программный алгоритм

```

Загрузим регистр нулевыми битами
Дополним хвостовую часть сообщения W нулевыми битами
While (пока еще есть необработанные биты)
Begin
    Сдвинем регистр на 1 бит влево и поместим очередной
    еще не обработанный бит из сообщения в 0 позицию
    регистра.
    If (из регистра был выдвинут бит со значением "1")
        Регистр = Регистр XOR Полином
End
Теперь в регистре содержится остаток.
  
```

Пример реализации расчёта CRC-16-CCITT

```

1  always_ff@(posedge clk or negedge nrst)
2  if(!nrst)
3      crc_reg    <= 16'hFFFF;
4  else if (init)
5      crc_reg    <= 16'hFFFF;
6  else begin
7      crc_reg[0] <= data_in ^ crc_reg[15];
8      crc_reg[1] <= crc_reg[0];
9      crc_reg[2] <= crc_reg[1];
10     crc_reg[3] <= crc_reg[2];
11     crc_reg[4] <= crc_reg[3];
12     crc_reg[5] <= crc_reg[4] ^ data_in ^ crc_reg[15];
13     crc_reg[6] <= crc_reg[5];
14     crc_reg[7] <= crc_reg[6];
15     crc_reg[8] <= crc_reg[7];
16     crc_reg[9] <= crc_reg[8];
17     crc_reg[10] <= crc_reg[9];
18     crc_reg[11] <= crc_reg[10];
19     crc_reg[12] <= crc_reg[11] ^ data_in ^ crc_reg[15];
20     crc_reg[13] <= crc_reg[12];
21     crc_reg[14] <= crc_reg[13];
22     crc_reg[15] <= crc_reg[14];
23 end
  
```

Методика выполнения лабораторной работы

Перечень управляющих сигналов модуля

Название	Направление	Разрядность, бит	Описание
clk	input	1	Тактовый сигнал.
nrst	input	1	Асинхронный сброс с низким активным уровнем.
init	input	1	Инициализировать регистр в начальное состояние. Ширина импульса – 1 такт.
start	input	1	Флаг начала расчёта CRC. Ширина импульса – 1 такт.
Data	input	32	Данные для расчёта CRC.
done	output	1	Флаг окончания расчёта CRC. Ширина импульса – 1 такт.
busy	output	1	Признак активного процесса расчёта CRC.
result	output	32	Результат расчёта CRC.

Варианты заданий

№	Название	Полином	Исходное значение	RefIn	RefOut	XorOut
1	CRC-32	0x04C11DB7	0xFFFFFFFF	Да	Да	0xFFFFFFFF
2	CRC-32/BZIP2	0x04C11DB7	0xFFFFFFFF	Нет	Нет	0xFFFFFFFF
3	CRC-32/JAMCRC	0x04C11DB7	0xFFFFFFFF	Да	Да	0x00000000
4	CRC-32/MPEG-2	0x04C11DB7	0xFFFFFFFF	Нет	Нет	0x00000000
5	CRC-32/POSIX	0x04C11DB7	0x00000000	Нет	Нет	0xFFFFFFFF
6	CRC-32/SATA	0x04C11DB7	0x52325032	Нет	Нет	0x00000000
7	CRC-32/XFER	0x000000AF	0x00000000	Нет	Нет	0x00000000
8	CRC-32C	0x1EDC6F41	0xFFFFFFFF	Да	Да	0xFFFFFFFF
9	CRC-32D	0xA833982B	0xFFFFFFFF	Да	Да	0xFFFFFFFF
10	CRC-32Q	0x814141AB	0x00000000	Нет	Нет	0x00000000

Лабораторное задание

В лабораторной работе необходимо в соответствии с вариантом реализовать устройство на HDL SystemVerilog и провести его функциональную верификацию



За основу желательно использовать модуль устройства, разработанного в Л/Р №1 и №2