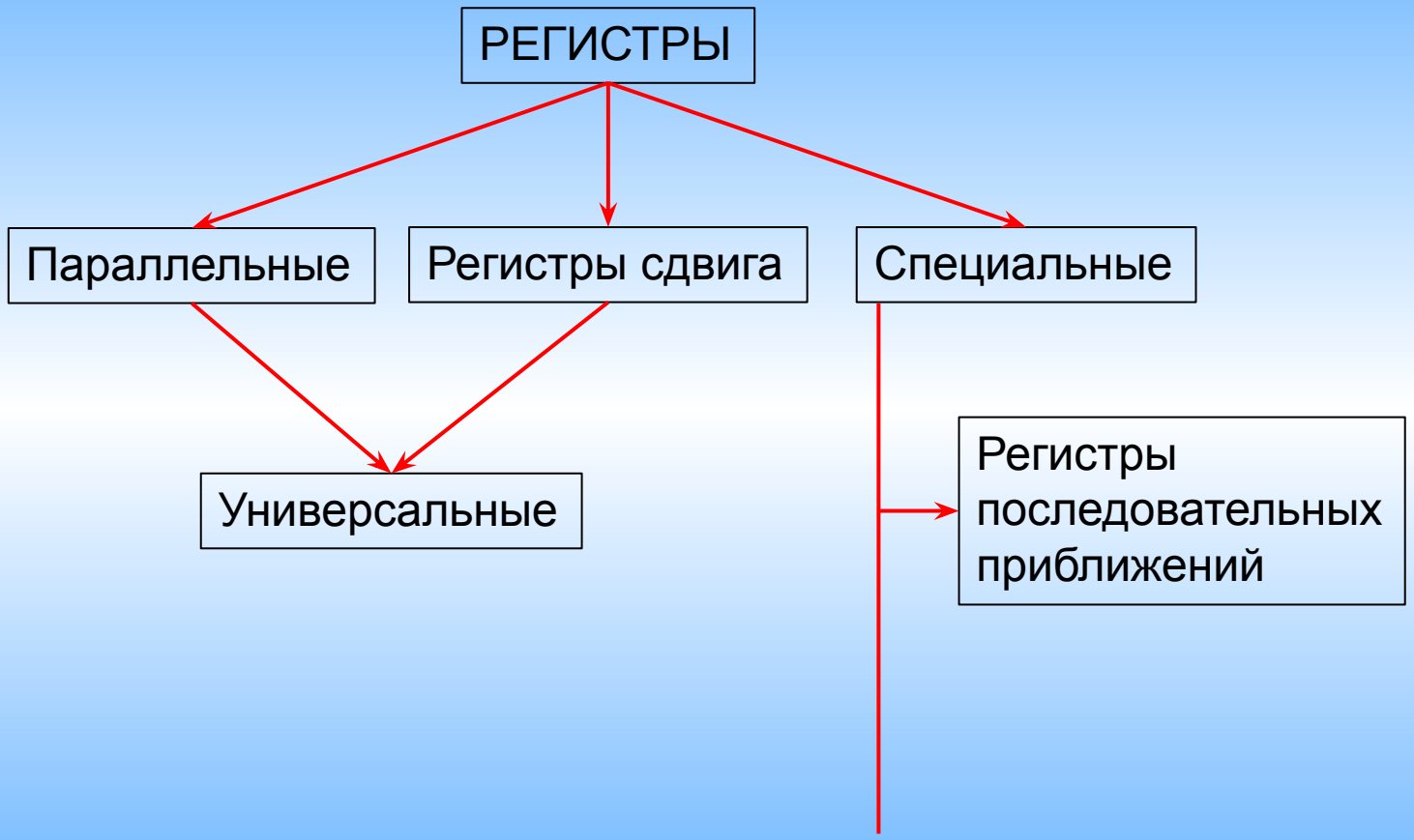
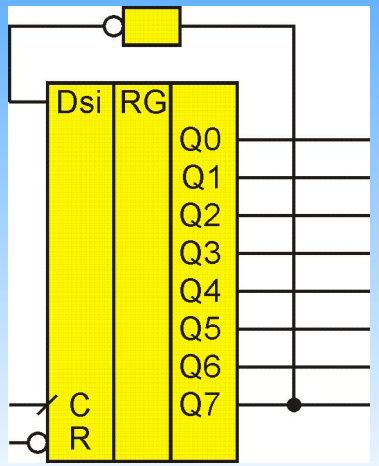
Регистры сдвига с обратными связями

Linear Feedback Shift Registers
LFSR

Классификация

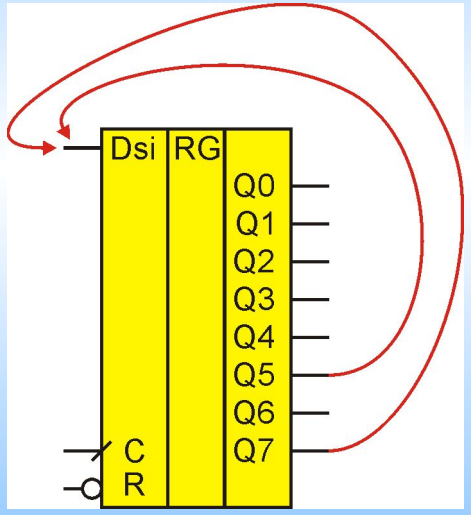


Обратные связи



Счетчик Джонсона
Упорядоченная последовательность

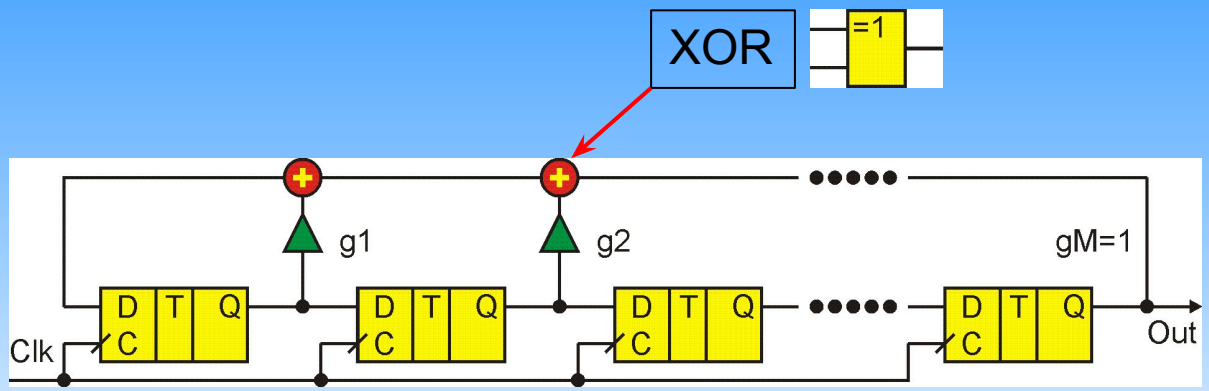
Крутой замес



Упорядоченная но очень сложная последовательность. Сложность зависит от длины регистра и расположения обратных связей.

Реализация обратных связей

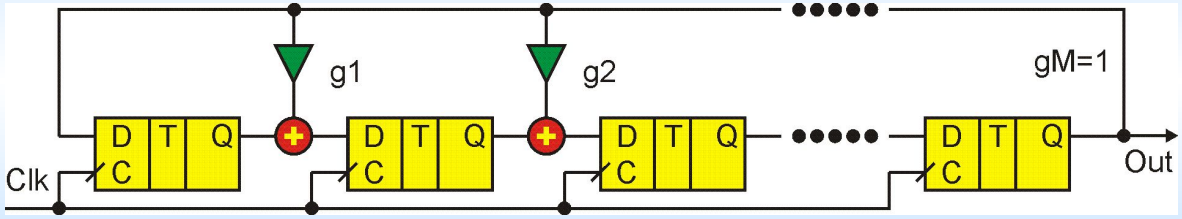
Конфигурация Фибоначчи



Начальное состояние всех триггеров не должно быть = 0

Конфигурация Галуа

Évariste Galois



Коэффициент.
Если g_i=1 – есть соединение и соответствующая функция XOR
Если g_i=0 – нет

Какие обратные связи сделать и зачем?

Последовательности максимальной длины

М-последовательности
Maximum length sequence

Число состояний набора из М триггеров = 2^M

Для нашего случая состояние со всеми нулями недопустимо!

Максимальное число допустимых состояний нашего регистра из М триггеров = $2^M - 1$

М	**2^M**	**Время перебора для частоты 100 МГц**
8	256	0,00000256 сек
16	65536	0,00065536 сек
32	4 294 967 296	42,949673 сек
64	1,84467E+19	5,85E+03 лет
127	1,70E+38	5,40E+22 лет

Это ВЕЧНОСТЬ

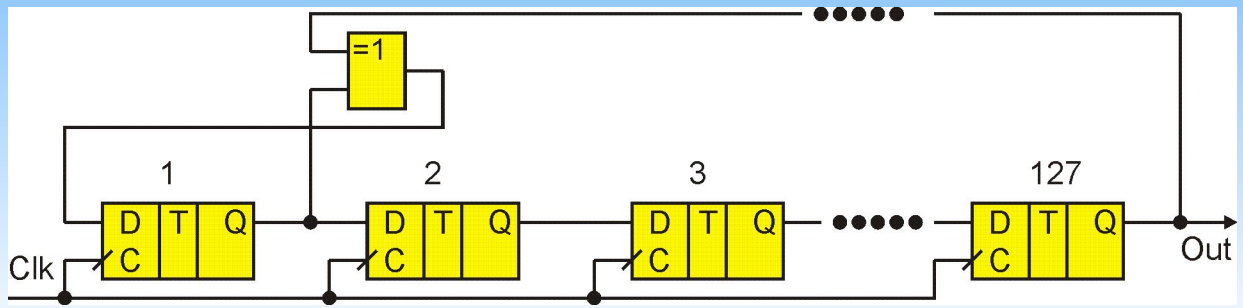
Справка:
Возраст Вселенной составляет 13,75E+9 лет

Последовательности максимальной длины

Как сделать М-последовательности

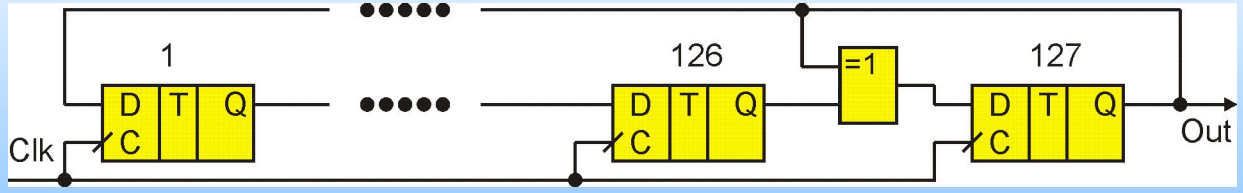
Реализация вечности

Фибоначчи



или

Галуа



Последовательности максимальной длины

Как сделать М-последовательности.

Кусок таблицы для конфигурации Галуа

N- количество триггеров

Схема с 2-мя отводами

Схема с 4-мя отводами

n	LFSR-2	LFSR-4	n	LFSR-2	LFSR-4	n	LFSR-2	LFSR-4
2	2, 1		24		24, 23, 21, 20	46		46, 40, 39, 38
3	3, 2		25	25, 22	25, 24, 23, 22	47	47, 42	47, 46, 43, 42
4	4, 3		26		26, 25, 24, 20	48		48, 44, 41, 39
5	5, 3	5, 4, 3, 2	27		27, 26, 25, 22	49	49, 40	49, 45, 44, 43
6	6, 5	6, 5, 3, 2	28	28, 25	28, 27, 24, 22	50		50, 48, 47, 46
7	7, 6	7, 6, 5, 4	29	29, 27	29, 28, 27, 25	51		51, 50, 48, 45
8		8, 6, 5, 4	30		30, 29, 26, 24	52	52, 49	52, 51, 49, 46
9	9, 5	9, 8, 6, 5	31	31, 28	31, 30, 29, 28	53		53, 52, 51, 47
10	10, 7	10, 9, 7, 6	32		32, 30, 26, 25	54		54, 51, 48, 46
11	11, 9	11, 10, 9, 7	33	33, 20	33, 32, 29, 27	55	55, 31	55, 54, 53, 49
12		12, 11, 8, 6	34		34, 31, 30, 26	56		56, 54, 52, 49
13		13, 12, 10, 9	35	35, 33	35, 34, 28, 27	57	57, 50	57, 55, 54, 52
14		14, 13, 11, 9	36	36, 25	36, 35, 29, 28	58	58, 39	58, 57, 53, 52
15	15, 14	15, 14, 13, 11	37		37, 36, 33, 31	59		59, 57, 55, 52
16		16, 14, 13, 11	38		38, 37, 33, 32	60	60, 59	60, 58, 56, 55
17	17, 14	17, 16, 15, 14	39	39, 35	39, 38, 35, 32	61		61, 60, 59, 56
18	18, 11	18, 17, 16, 13	40		40, 37, 36, 35	62		62, 59, 57, 56
19		19, 18, 17, 14	41	41, 38	41, 40, 39, 38	63	63, 62	63, 62, 59, 58
20	20, 17	20, 19, 16, 14	42		42, 40, 37, 35	64		64, 63, 61, 60
21	21, 19	21, 20, 19, 16	43		43, 42, 38, 37	65	65, 47	65, 64, 62, 61
22	22, 21	22, 19, 18, 17	44		44, 42, 39, 38	66		66, 60, 58, 57
23	23, 18	23, 22, 20, 18	45		45, 44, 42, 41	67		67, 66, 65, 62

Переход от номеров отводов Галуа к Фибоначчи и наоборот:

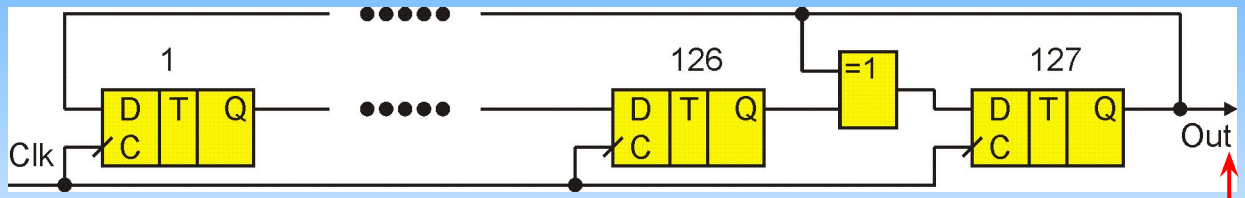
M-g

(кроме старшего).

[20, 19, 16, 14] ⇔ [20, 1, 4, 6]

Псевдослучайная последовательность. PRBS

Pseudo Random Binary Sequence



Последовательность детерминирована, но никогда не повторится. Период 5,40E+22 лет При частоте Clk 100 МГц.

Такая последовательность называется псевдослучайной
или
ПСП
Pseudo Random Binary Sequence
PRBS

Защита информации

Идея

Случайная последовательность

Исходные данные

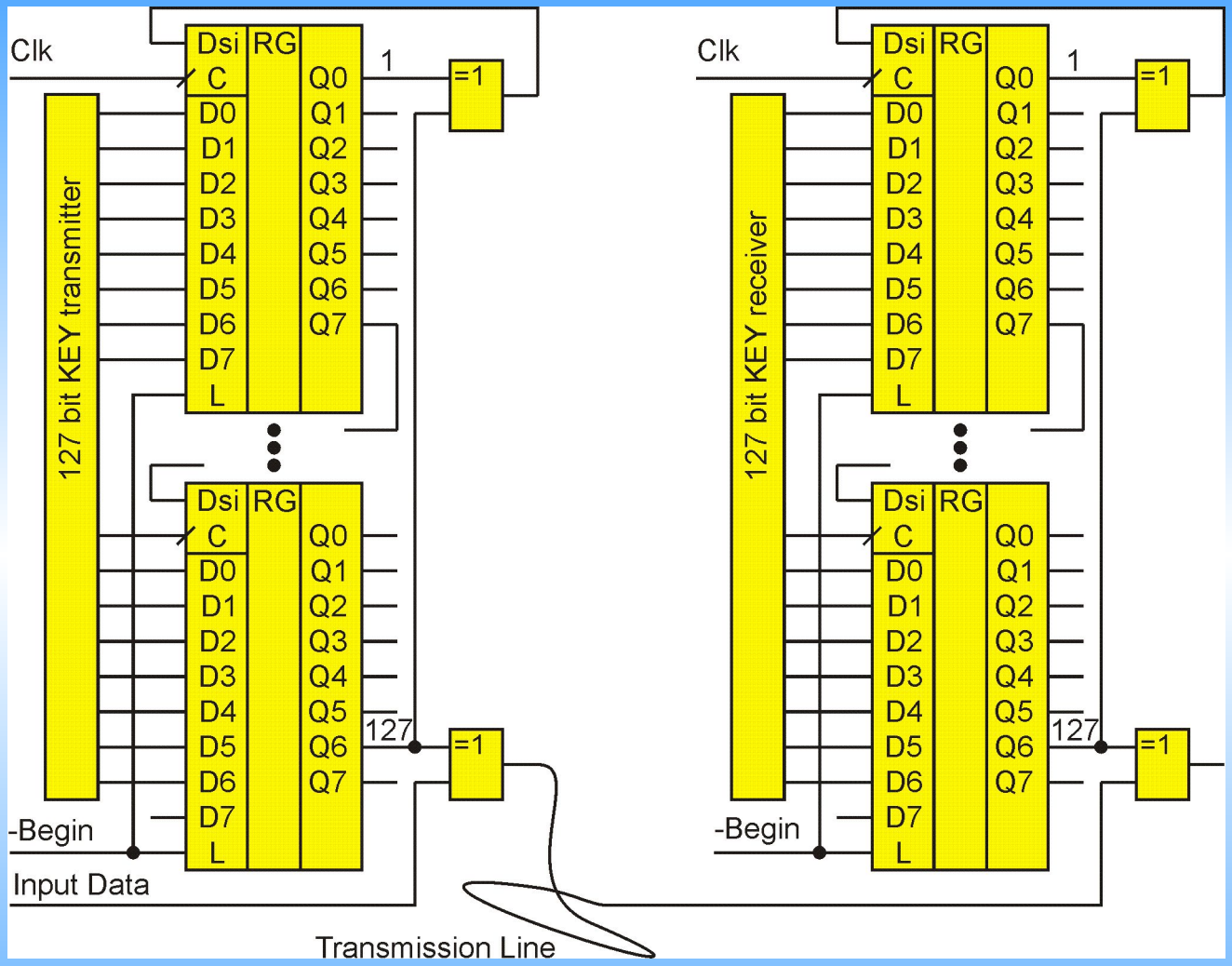
Здесь полная мешанина

Восстановленные данные

PRBS	D	Q
0	0	0
0	1	1
1	0	0
1	1	1

Q=Data

Защита информации



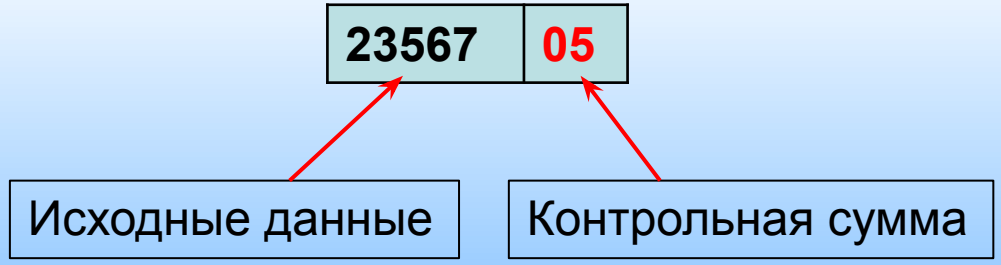
Ключ приемника должен совпадать с ключом приемника.
Используется детерминированность ПСП.

Проверка целостности информации

Контрольная сумма или циклический избыточный код
Cyclic Redundancy Code (CRC)

ИДЕЯ МЕТОДА

- Допустим надо передать (или сохранить) десятичное число 23567.
- Возможно искажение. Как проверить правильность числа?
- Поделить исходное число на какую либо постоянную. Допустим на 23. В результате целочисленного деления получим 1024 и остаток 15. Если теперь к исходному числу прибавить 15, то новое число будет делиться на 23 без остатка. Это признак правильности. Но исходное число будет потеряно.
- Выход. Число 23567 дополняем нулями. Количество нулей должно совпадать с разрядностью остатка (делителя). Получаем 2356700. Теперь это число делим на 23. Получаем 102465 и остаток 05. Интересует только остаток. Храним или передаем число в форме 2356705.



- При приеме делим 2356705 на наше 23.
  - ✔ Если остаток ≠ 0, то число искажено.
  - ✔ Если остаток = 0, то число с некоторой вероятностью правильное.

Проверка целостности информации

Двоичная информация

- Любая информация представляет собой набор 0 и 1.
  10010001011110001010100100100101010010001111010
- В соответствии с идеей надо дописать слово нулями
  10010001011110001010100100100101010010001111010**0000**
- взять некий делитель, например 1011(с разрядностью по количеству дописанных нулей.
- Поделить
- Дописать вместо нулей остаток от деления.
  10010001011110001010100100100101010010001111010**xxxx**
- Хранить или передавать в такой форме.
- Перед использование проверить на делимость без остатка на 1011.

Все по идее просто, но
Делить – это долго и муторно!!!
Даже для этого примера лень было найти реальный остаток и написал xxxx

Нужна другая БЫСТРАЯ арифметика.

Контрольная сумма

Представление битовых последовательностей полиномами.

bit N	9	8	7	6	5	4	3	2	1	0
	1	0	0	1	1	0	1	0	1	1
	X^9			X^6	X^5		X^3		X^1	1

$$1001101011=X^9+X^6+X^5+X^3+X^1+1$$

Контрольная сумма

Полиномиальная арифметика.
Арифметика по модулю 2 (без переносов).

XOR

$$0+0=0 \qquad 0-0=0$$
$$1+0=1 \qquad 1-0=1$$
$$0+1=1 \qquad 0-1=1$$
$$1+1=0 \qquad 1-1=0$$

Суммирование = вычитанию

$A=X^9+X^6+X^5+X^3+X^1+1$ $\qquad$ $B=X^6+X^3+X^2+1$

$A+B=A-B=B-A=\ X^9+X^6+X^5+X^3+X^1+1+\ X^6+X^3+X^2+1=\ X^9+X^5+X^2+X^1$

Переноса нет!

Bit N	9	8	7	6	5	4	3	2	1	0
A	1	0	0	1	1	0	1	0	1	1
B	0	0	0	1	0	0	1	1	0	1
SUM	1	0	0	0	1	0	0	1	1	0
	X^9				X^5			X^2	X^1	

Умножение

Полиномиальная арифметика.
Арифметика по модулю 2

$A=X^9+X^6+X^5+X^3+X^1+1$ $B=X^6+X^3+X^2+1$

$AxB=(X^9+X^6+X^5+X^3+X^1+1)x(X^6+X^3+X^2+1)=$
$X^{15}+X^{12}+X^{11}+X^9+X^7+X^6+ X^{12}+X^9+X^8+X^6+X^4+X^3+ X^{12}+X^8+X^7+X^5+X^3+X^2+$
$X^9+X^6+X^5+X^3+X^1+1 =$

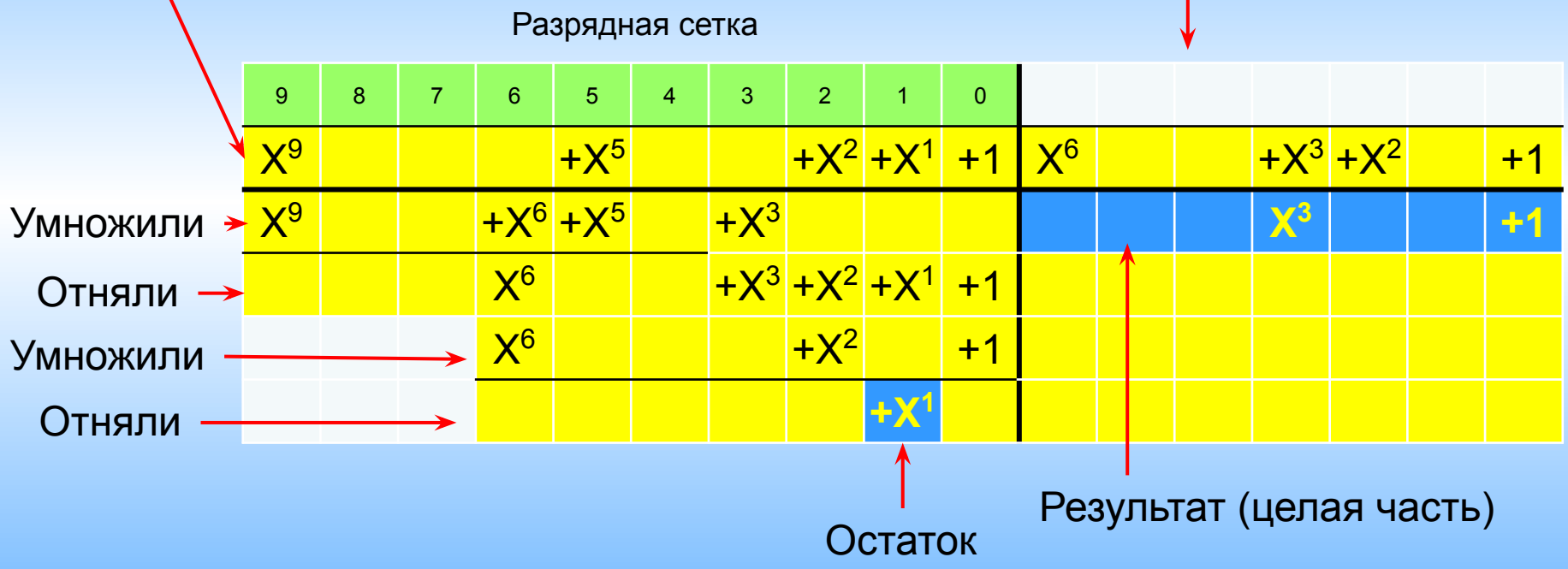
$X^{15}+X^{12}+X^{11}+X^9+X^6+X^4+X^3+X^2+X^1+1$

Деление

Полиномиальная арифметика.
Арифметика по модулю 2

$A=X^9+X^6+X^5+X^3+X^1+1$

$B=X^6+X^3+X^2+1$



Это гораздо проще чем обычное деление.

Проверка целостности информации

Получение остатка с помощью LFSR

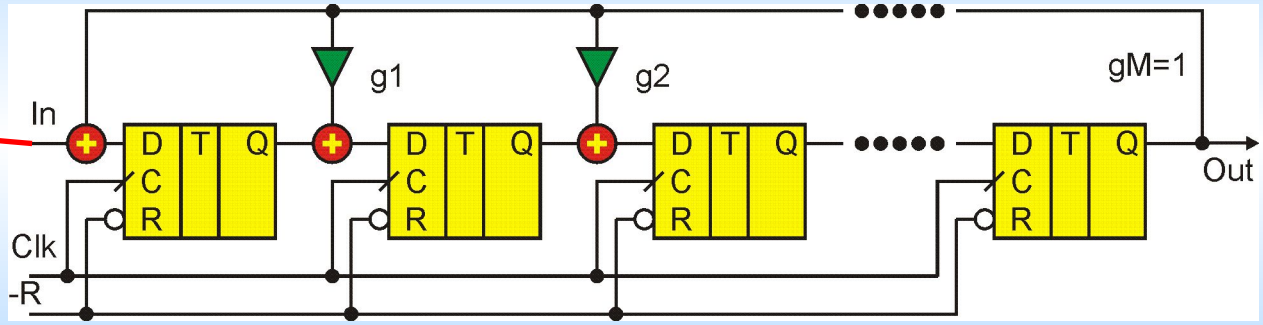
N разрядное входной полином

Дописанное M нулями поле для CRC

10010001011110001010100100100101010010001111010100..0

Все слово подается поразрядно на вход LFSR. Старшим разрядом вперед.

LFSR со входом



За N+M тактов в триггерах останется остаток от деления по модулю 2.

Такие схемы просто реализуется как на аппаратном уровне, так и на программном.

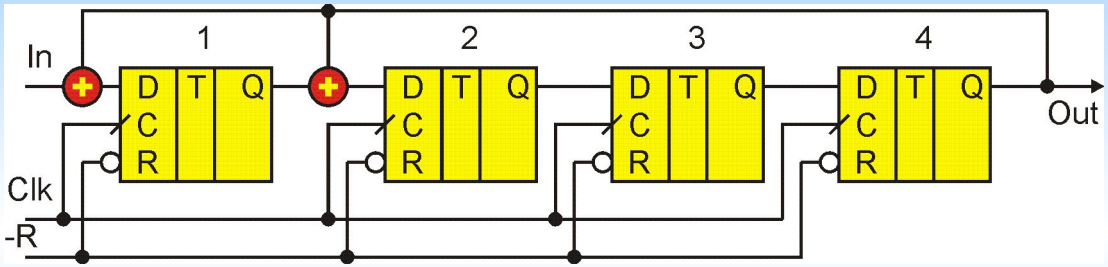
Количество триггеров соответствует полиному – делителю.

Делитель называется порождающим полиномом.

Контрольная сумма

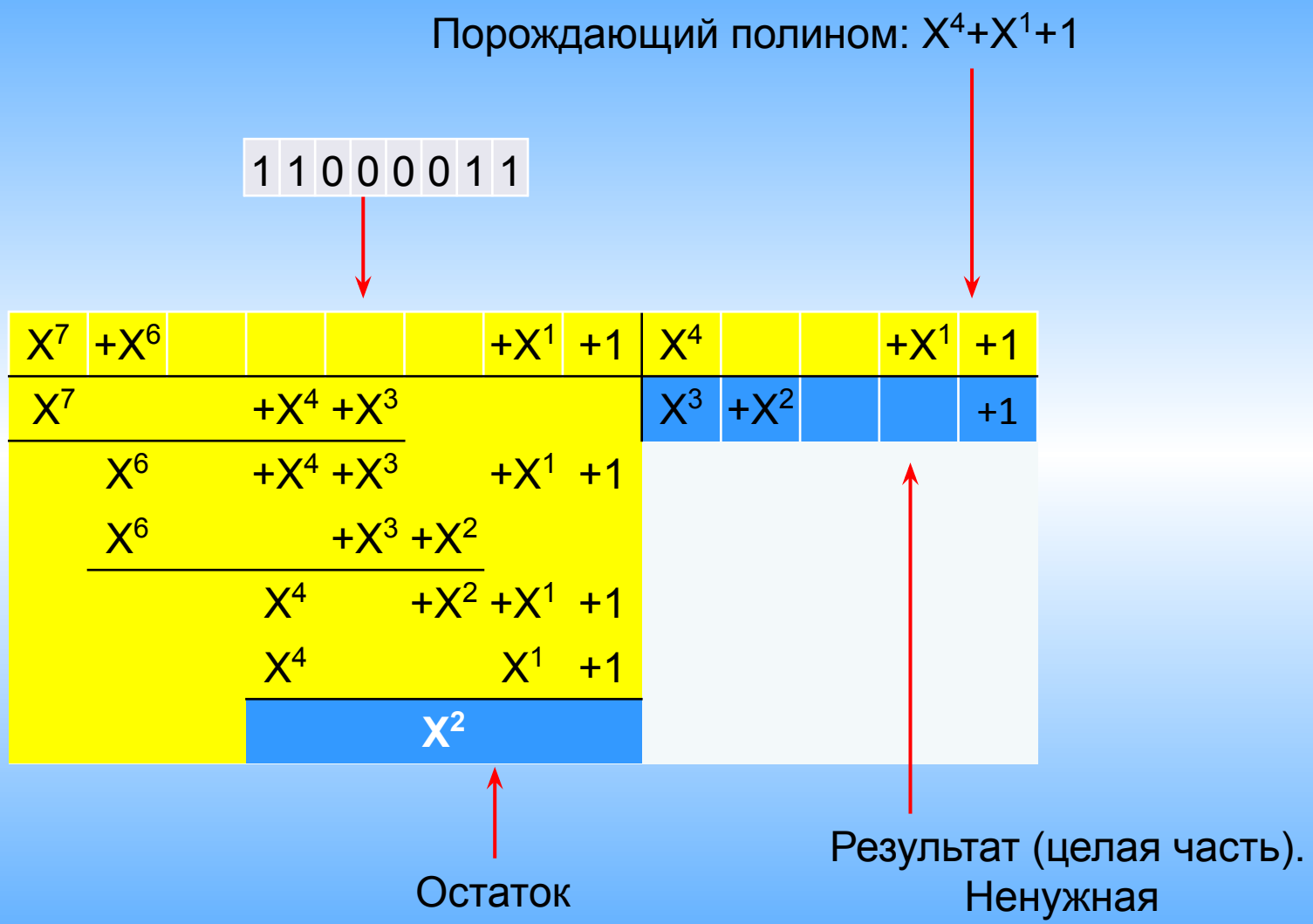
Получение остатка с помощью LFSR. Пример.

Порождающий полином: X^4+X^1+1



Контрольная сумма

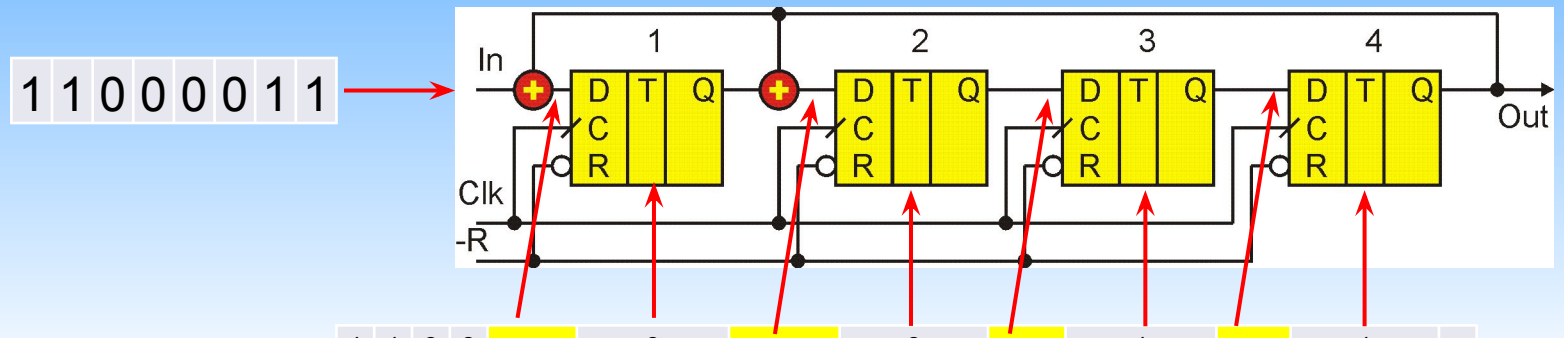
Пример. Кодирование.



Контрольная сумма

Пример. Кодирование.

Порождающий полином: X^4+X^1+1



1	1	0	0		0		0		1		1	4t
				1		1		0		1		
	1	1	0		1		1		0		1	5t
				1		0		1		0		
		1	1		1		0		1		0	6t
				1		1		0		1		
			1		1		1		0		1	7t
				0		0		1		0		
					0		0		1		0	8t

Младший разряд +1 | Разряд X^1 | Разряд X^2 | Старший разряд

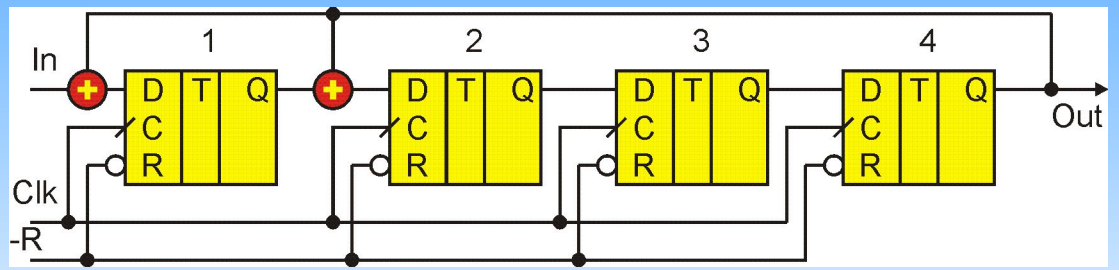
Остаток X^2

Контрольная сумма

Пример. Кодирование.

Порождающий полином: X^4+X^1+1

Старшим разрядом вперед. →



1	X^1					X^6	X^7

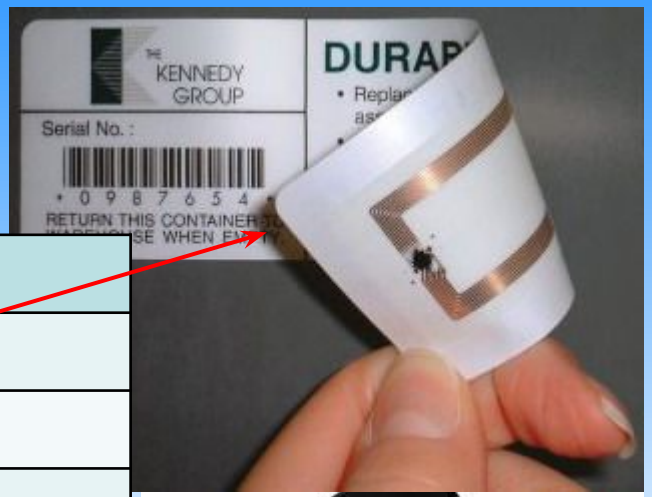
X^7	$+X^6$					$+X^1$	$+1$	X^4			$+X^1$	$+1$
X^7			$+X^4$	$+X^3$				X^3	$+X^2$	0		$+1$
	X^6		$+X^4$	$+X^3$		$+X^1$	$+1$					
	X^6			$+X^3$	$+X^2$							
		0	X^4		$+X^2$	$+X^1$	$+1$					
		0	0	0	0	0	0					
			X^4		$+X^2$	$+X^1$	$+1$					
			X^4			X^1	$+1$					
				X^2								

1	X^1	X^2	X^3	X^4	X^5	X^6	X^7	
1	1	0	0	0	0	1	1	4t
	1	X^1	X^2	X^3	X^4	X^5	X^6	
	1	1	0	1	1	0	1	5t
		1	X^1	X^2	X^3	X^4	X^5	
		1	1	1	0	1	0	6t
			1	X^1	X^2	X^3	X^4	
			1	1	1	0	1	7t
				1	X^1	X^2	X^3	
				0	0	1	0	8t

CRC

Cyclic Redundancy Code

CRC-4-ITU	$x^4 + x + 1$	ITU-T G.704 (E1)
CRC-5-EPC	$x^5 + x^3 + 1$	Gen 2 RFID
CRC-7	$x^7 + x^3 + 1$	ITU-T G.707, ITU-T G.832, MMC, SD
CRC-8-CCITT	$x^8 + x^2 + x + 1$	ATM HEC, ISDN ITU-T I.432.1
CRC-8-Dallas/Maxim	$x^8 + x^5 + x4 + 1$	1-Wire bus
CRC-16-IBM *CRC-16-ANSI*	$x^{16} + x^{15} + x^2 + 1$	Bisync, Modbus, USB, ANSI X3.28 …
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$	X.25, V.41, HDLC, XMODEM, Bluetooth, …
CRC-32-IEEE 802.3	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	Ethernet, Wi-Fi, ……..



Контрольная сумма

Содержание

- Обратные связи
- Конфигурация Фибоначчи
- Конфигурация Галуа
- М-последовательности
- Переход от номеров отводов Галуа к Фибоначчи и наоборот
- Pseudo Random Binary Sequence
- Защита информации
  - Ключ приемника должен совпадать с ключом приемника
- Контрольная сумма или циклический избыточный код
  - Идея метода
  - Представление битовых последовательностей полиномами
  - Полиномиальная арифметика -арифметика по модулю 2 (без переносов)
  - Сложение = вычитание
  - Умножение
  - Деление
  - Получение остатка с помощью LFSR
  - Получение остатка с помощью LFSR. Пример.
  - Примеры CRC. E1, RFID, SD, 1-Wire, Bluetooth, Ethernet, Wi-Fi …