Лабораторная работа №2
Контроль целостности (биты четности, контрольные цифры, CRC и ECC)
В лабораторной работе необходимо определить контрольные данные с использованием следующих способов:

- битов четности. В качестве исходных данных принять битовое представление букв фамилии в соответствии с кодировкой Windows 1251 (табл.6.2);

Буква	Битовая строка	Паритетный бит
четный (odd)	нечетный (even)	

- контрольных цифр. В качестве исходных данных принять необходимое количество цифр (за исключением контрольной) из строки, состоящей из кодов букв фамилии, имени и отчества согласно их положению в алфавите:

- по алгоритму Луна (15 цифр);

- для штрихкода по стандарту EAN-13 (12 цифр);

- для ИНН физического лица (10 цифр);

- для кодов станций на железнодорожном транспорте (5 цифр);
- контрольных сумм (CRC). В качестве исходных данных принять коды 1-ой, 2-ой и 3-ей буквы своей фамилии согласно их положению в алфавите; порождающего полинома - G(x) = x4 + x1 + x0.

- кода коррекции ошибок (ECC). В качестве исходных данных принять первые 11 битов первых двух буквы своей фамилии в соответствии с кодировкой Windows 1251 (табл.6.2). Рассчитать вектор контрольных битов и вектора синдромов при отсутствии ошибки, одиночной и двойной ошибке.

При оформлении отчета необходимо привести необходимые таблицы, исходные данные, расчеты и результаты.

ПРОТОКОЛЫ КОНТРОЛЯ ЦЕЛОСТНОСТИ

15.1. Общие сведения.

15.2. Проверка четности.

15.3. Использование контрольных цифр.

15.4. Использование контрольных сумм.

15.5. Использование ECC.

15.6. Использование ЭЦП.

15.7. Использование MAC-кодов.

15.8. Комбинированные методы (на примере жестких магнитных дисков).

Вопросы для самопроверки.

15.1. Общие сведения

Как было отмечено в первой лекции, целостность является одним из трех ключевых свойств информации (доступность, целостность и конфиденциальность). При этом
под **целостностью** понимается свойство информации, характеризующее ее устойчивость к случайному или преднамеренному разрушению или несанкционированному изменению.

Рассмотрим некоторые способы и методы контроля целостности.

15.2. Проверка четности

Представляет собой самый простой способ обеспечения целостности при хранении или передаче данных. Битовая строка (обычно длиной 7-8 бит), контроль которой необходимо выполнить, дополняется одним, так называемым **паритетным битом** (англ. parity bit). Существует две разновидности проверки четности: с **четным (odd)** и **нечетным (even)** паритетным битом. В первом случае при записи или пересылке данных паритетный бит устанавливается равным 1, если количество единиц в контролируемой строке четное, и 0 – если нечетное. В случае нечетного паритетного бита поступают наоборот.

Таблица 15.1

Примеры установки бита четности

Битовая строка	Паритетный бит	
	четный (odd)	нечетный (even)
1100 1011	0	1
1001 1001	1	0
1111 1111	1	0
0000 0000	0	1

Недостатки:

- исправление ошибки невозможно;

- в случае изменения состояния четного количества бит (например, двух), вычисленный паритетный бит совпадет с записанным. Т. е. ошибка не будет обнаружена. В тоже время, согласно статистики, приблизительно 90% всех ошибок памяти происходит именно с одиночным разрядом. Т. о. проверки четности бывает достаточно для большинства ситуаций.

15.3. Использование контрольных цифр

В отличие от предыдущего способа для контроля целостности используется не бит, а цифра. Обычно, контролируемый набор цифр вначале по определенным правилам складывается, а затем берется остаток от деления по модулю, который и является контрольной цифрой. Ниже рассматриваются некоторые системы кодирования с использованием контрольной цифры:

- алгоритм Луна;

- штрихкод по стандарту EAN-13;

- заграничный паспорт гражданина РФ с биометрическими данными;

- индивидуальный номер налогоплательщика;

- коды станций на железнодорожном транспорте.

Алгоритм Луна (англ. Luhn algorithm) - алгоритм вычисления контрольной цифры в соответствии со стандартом ISO/IEC 7812 «Идентификационные карты. Идентификация эмитентов». Алгоритм разработан сотрудником фирмы IBM Гансом Питером Луном в 1954 г. Используется для подсчета контрольной цифры:

- номеров всех банковских карт;

- номеров некоторых дисконтных карт;

- кодов полисов обязательного медицинского страхования;

- единого 8-значного номера железнодорожного вагона на РЖД;

- IMEI-кодов (англ. International Mobile Equipment Identity - международный идентификатор мобильного оборудования);

- ICCID-кодов (англ. Integrated Circuit Card ID - идентификатор карты с интегрированной микросхемой);

- т. д.

а) банковская карта

б) полис медицинского страхования

в) цистерна

г) мобильный телефон (IMEI)

д) SIM-карта (ICCID)

Рис.15.1. Номера и коды с контрольными цифрами

В следующей таблице приведен порядок вычисления контрольной цифры на примере кода полиса медицинского страхования (рис. 15.1б).

Таблица 15.2

Вычисление контрольной цифры по алгоритму Луна

(если количество цифр в коде четное)

№ п/п	Описание операции	Пример
1	Каждая из цифр, стоящая в нечетной позиции, умножается на 2, после чего вычисляется остаток от деления на 9.	(2 * 2) mod 9 = 4 (5 * 2) mod 9 = 1 (6 * 2) mod 9 = 3 (0 * 2) mod 9 = 0 (4 * 2) mod 9 = 8 (0 * 2) mod 9 = 0 (0 * 2) mod 9 = 0 (1 * 2) mod 9 = 2
2	Вычисляется сумма остатков $S_н$.	$S_н$ = 4 + 1 + 3 + 0 + 8 + 0 + 0 + 2 = 18
3	Вычисляется сумма цифр $S_ч$, стоящих в четных позициях, за исключением последней.	$S_ч$ = 7 + 8 + 2 + 8 + 2 + 0 + 2 = 29
4	Вычисляется контрольная (последняя) цифра **cd** из уравнения ($S_н$ + $S_ч$ + cd) mod 10 = 0.	cd = 3 (18 + 29 + 3) mod 10 = 0

Если количество цифр в коде нечетное (например, для IMEI-кодов), то 1 и 2 операция выполняются для цифр, стоящих в четных позициях, 3 операция – для цифр, стоящих в нечетных позициях.

Штрихкод по стандарту EAN-13 - одна из вариаций Европейского стандарта штрихкода, предназначенного для кодирования идентификатора товара и производителя. Регламентируется ГОСТ ИСО/МЭК 15420-2001 «Автоматическая идентификация. Кодирование штриховое. Спецификация символики EAN/UPC (ЕАН/ЮПиСи)».

Рис.15.2. Штрихкод EAN-13

В следующей таблице приведен порядок вычисления контрольной цифры по стандарту EAN-13.

Таблица 15.3

№ п/п	Описание операции	Пример
1	Вычисляется сумма цифр $S_н$, стоящих в нечетных позициях, за исключением последней.	$S_н = 5 + 0 + 2 + 4 + 2 + 4 = 17$
2	Вычисляется утроенная сумма цифр $S_ч$, стоящих в четных позициях.	$S_ч = 3 * (9 + 1 + 3 + 1 + 3 + 5) = 66$
3	Вычисляется контрольная (последняя) цифра **cd** из уравнения $(S_н + S_ч + cd) \bmod 10 = 0$.	cd = 7 $(17 + 66 + 7) \bmod 10 = 0$

В России с 2009 г. во всех субъектах РФ действуют пункты выдачи паспортно-визовых документов нового поколения - **заграничных паспортов гражданина РФ с биометрическими данными.**
В документе используются различные способы защиты, в т. ч. защита целостности за счет контрольных цифр. В пластиковой странице с фотографией владельца и встроенным внутри чипом имеется т. н. машиносчитываемая зона (МСЗ).

Рис.15.3. Машиносчитываемая зона паспорта

(синим контуром выделены контрольные цифры)

В нижней строке МСЗ используются пять контрольных цифр. Ее структура приведена в следующей таблице.

Таблица 15.4

Структура нижней строки МСЗ

Контрольная цифра	Позиции знака, используемые для расчета контрольных цифр	Позиция контрольной цифры
Номер паспорта	1-9	10
Дата рождения	14-19	20
Дата истечения срока действия паспорта	22-27	28

Личный номер	29-42	43
Заключительная контрольная цифра	1-10, 14-20, 22-43 (позиции 11-13 и 23 исключаются из расчета)	44

Алгоритм расчета контрольных цифр заключается в перемножении каждой цифры соответствующего элемента данных на весовой показатель повторяющейся функции "731 731 ...", суммировании полученных произведений и взятии остатка от деления на 10. Если в элементе данных встречаются буквы латинского алфавита, то при расчете они заменяются на числа от 10 (A) до 35 (Z); знак "<" соответствует 0. Ниже приведен пример расчета контрольных цифр.

Таблица 15.5

Пример расчета контрольных цифр нижней строки МСЗ

Назначение	Номер паспорта	КЦ		Дата рождения	КЦ		Дата истечения срока действия	КЦ	Личный номер	КЦ	КЦ																						
№ позиции	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	...	42	43	44
Нижняя строка	7	2	5	4	3	6	2	3	9	**5**	R	U	S	7	3	0	1	0	7	**2**	M	2	3	0	7	1	2	**7**	<	...	<	**0**	**2**

МСЗ							
Весовой показатель	7	3 1 7	3 1 7	3 1	7	7 3 1 7 3 1 7	7 3 1 7 3 1 7 7 ... 3 1
Расчет контрольной цифры в элементе данных	(7*7+2*3+5*1+4*7+3*3+6*1+2*7+3*3+9*1) mod 10 = 135 mod 10 = **5**	(7*7+3*3+0*1+1*7+0*3+7*1) mod 10 = 72 mod 10 = **2**	(2*7+3*3+0*1+7*7+1*3+2*1) mod 10 = 77 mod 10 = **7**	(0*7+…+0*3) mod 10 = **0**			
Расчет заключительной контрольной цифры	[(135 + 5*7) + (72 + 2*7) + (77 + 7*7) + (0 + 0*1)] mod 10 = 382 mod 10 = **2**						

Индивидуальный номер налогоплательщика (ИНН) - уникальный идентификатор, присваиваемый юридическому или физическому лицу для учета уплаты налогов в Российской Федерации. При постановке на налоговый учет подотчетному лицу выдается свидетельство, в котором указывается его

ИНН.

Министерство Российской Федерации по налогам и сборам

СВИДЕТЕЛЬСТВО

о постановке на учет в налоговом органе физического лица по месту жительства на территории Российской Федерации

Настоящее Свидетельство выдано в соответствии с положением части первой Налогового кодекса Российской Федерации, принятого Федеральным законом от 31 июля 1998 года N146-ФЗ, физическому лицу [redacted]
(фамилия, имя, отчество)

пол муж.

дата рождения 7 января 1973 года
(число, месяц, год)

место рождения БЕЛОГОРСК Г,,,643
(указывается в точном соответствии с записью в документе, удостоверяющем личность)

и подтверждает постановку физического лица на учет 22 марта 2000 года
(число,месяц,год постановки на учет)

В ИНСПЕКЦИИ МНС РОССИИ ПО ЖЕЛЕЗНОДОРОЖНОМУ РАЙОНУ г.ХАБАРОВСКА | 2 | 7 | 2 | 4 |
(наименование государственной налоговой инспекции и ее код)

Идентификационный номер налогоплательщика (ИНН) | 2 | 7 | 2 | 4 | 0 | 7 | 0 | 3 | 1 | 7 | 9 | 0 |

Дата выдачи Свидетельства 22 марта 2000 года
(число,месяц,год)

Свидетельство применяется во всех предусмотренных законодательством случаях и предъявляется вместе с документом, удостоверяющим личность физического лица и место его жительства на территории Российской Федерации .

Свидетельство подлежит замене в случае переезда физического лица на новое место жительства на территорию, подведомственную другой государственной налоговой инспекции, изменения приведенных в нем сведений, а также в случае порчи, утери.

Руководитель инспекции МНС России по Железнодорожному району г.Хабаровска — Л.Г.Бурковская
(подпись, фамилия, имя, отчество)

МП

серия 27 № 0042973

Рис.15.4. Свидетельство о постановке на учет

Контрольная (контрольные) цифра ИНН определяется по следующим формулам:

- для десятизначного ИНН юридического лица:

n10 = ((2n1 + 4n2 + 10n3 + 3n4 + 5n5 + 9n6 + 4n7 + 6n8 + 8n9) mod 11) mod 10; (15.1)

- для двенадцатизначного ИНН физического лица:

n11 = ((7n1 + 2n2 + 4n3 + 10n4 + 3n5 + 5n6 + 9n7 + 4n8 + 6n9 + 8n10) mod 11) mod 10, (15.2)

n12 = ((3n1 + 7n2 + 2n3 + 4n4 + 10n5 + 3n6 + 5n7 + 9n8 + 4n9 + 6n10 + 8n11) mod 11) mod 10, (15.3)

где ni - i-ая цифра ИНН.

Для ИНН физического лица, отображенного на рис. 15.4, контрольные цифры:

n11 = ((7*2 + 2*7 + 4*2 + 10*4 + 3*0 + 5*7 + 9*0 + 4*3 + 6*1 + 8*7) mod 11) mod 10 = (185 mod 11) mod 10 = 9 mod 10 = 9,

n12 = ((3*2 + 7*7 + 2*2 + 4*4 + 10*0 + 3*7 + 5*0 + 9*3 + 4*1 + 6*7 + 8*9) mod 11) mod 10 = (241 mod 11) mod 10 = 10 mod 10 = 0.

Коды станций на железнодорожном транспорте. В информационных системах железнодорожного транспорта приняты различные способы кодирования станций. В **АСУЖТ** используется код станции, состоящий из 6 цифр (n1n2n3n4n5n6). Последняя цифра кода (n6) является контрольной и определяется по следующей формуле:

n6 = (1n1 + 2n2 + 3n3 + 4n4 + 5n5) mod 11. (15.4)

Если остаток от деления меньше 10, то он является контрольной цифрой, иначе выполняют сдвиг весового ряда на две позиции и вычисления повторяют:

n6 = (3n1 + 4n2 + 5n3 + 6n4 + 7n5) mod 11. (15.5)

Если новый остаток от деления вновь получится равным 10, то контрольная цифра принимается равной 0, иначе - остатку, вычисленному по формуле 15.5.

Первые четыре цифры АСУЖТ для станций, открытых для грузовых операций, называют кодом **Единой сетевой разметки (ЕСР)**. Вариация кода ЕСР с контрольной цифрой состоит из 5 знаков (n1n2n3n4n5), последний из которых (n5) определяется точно также, как и для кода станции в АСУЖТ. Отличие заключается в использовании сокращенных весовых рядов (1, 2, 3, 4) и (3, 4, 5, 6). Т. к. пятая цифра для грузовых станций в АСУЖТ принимается равной 0, то контрольные цифры кодов станций АСУЖТ и ЕСР совпадают. В частности, код станции Хабаровск-1 Дальневосточной железной дороги:

- АСУЖТ: код - 970406, контрольная цифра - n6 = (1*9 + 7*2 + 0*3 + 4*4 + 5*0) mod 11 = 39 mod 11 = 6;

- ЕСР: код - 97046, контрольная цифра - n5 = (1*9 + 7*2 + 0*3 + 4*4) mod 11 = 39 mod 11 = 6.

15.4. Использование контрольных сумм

Контрольные суммы (checksums или CRC) являются более надежным способом обеспечения целостности, чем биты четности или контрольные цифры. В англоязычной литературе CRC расшифровывается двояко в зависимости от контекста: Cyclic Redundancy Code или Cyclic Redundancy Check. Под первой расшифровкой понимают циклический код, под второй – хеш-образ.
Циклические коды основаны на полиноминальной арифметике по модулю 2 (полиноминальном делении без переноса). Вместо представления делимого (исходного сообщения, входных данных), делителя (порождающего полинома), частного (целой части) и остатка (контрольной суммы, CRC) в виде положительных целых чисел, их можно представить в виде полиномов с двоичными коэффициентами или в виде строки бит, каждый из которых является коэффициентом полинома. Например, десятичное число 1910 в двоичной системе счисления имеет вид 100112, что совпадает с полиномом

1*x4 + 0*x3 + 0*x2 + 1*x1 + 1*x0 = x4 + x1 + x0. (15.6)

Значение контрольной суммы с порождающим полиномом G(x) определяется по формуле:

R(x) = P(x) * xN mod G(x), (15.7)

где R(x) - полином, представляющий значение контрольной суммы;

P(x) - полином, представляющий входные данные;

G(x) - порождающий полином;

N - максимальная степень порождающего полинома.

Умножение xN эквивалентно приписыванию N нулевых битов к входным данным. Полиноминальное

деление без переноса выполняется по следующим правилам:

- при наличии у промежуточного остатка в качестве старшего бита «1», он складывается по модулю 2 (XOR, исключающее ИЛИ) с битовым представление порождающего полинома и в частное записывается «1»;

- в противном случае выполняется сложение по модулю 2 промежуточного остатка с нулевой битовой строкой длиной N+1 и в частное записывается «0».

В следующей таблице приведены примеры определения контрольных сумм для порождающего полинома G(x) = x4 + x1 + x0 (дели, 1910; N = 4; xN = 100002).

Таблица 15.6

Примеры определения контрольных сумм

Делимое P(x) (входные данные)	101112 (2310)	100112 (1910)	100012 (1710)
P(x) * xN	1011100002 (36810)	1001100002 (30410)	1000100002 (27210)
Деление P(x) * xN mod G(x)	101110000 10011 1 01000 00000 0 10000 10011 1 00110 00000 0 01100 00000 0 1100	100110000 10011 1 00000 00000 0 00000 00000 0 00000 00000 0 00000 00000 0 0000	100010000 10011 1 00100 00000 0 01000 00000 0 10000 10011 1 00110 00000 0 0110
Частное	101002 (2010)	100002 (1610)	100102 (1810)

Остаток R(x) (контрольная сумма)	11002 (1210)	00002 (010)	01102 (610)
Входные данные с контрольной суммой	10111 11002 (38010)	10011 00002 (30410)	10001 01102 (27810)

Принимающая сторона для проверки целостности полученных данных может сделать одно из следующих равноценных действий:

- выделить входные данные, вычислить для них контрольную сумму (не забыв при этом дополнить данные N нулевыми битами) и сравнить ее с переданной;

- поделить входные данные с контрольной суммой (последняя строка табл. 15.6) на делитель, представляющий порождающий полином G(x). В результате должен получиться нулевой остаток.

Как было отмечено выше, использование циклических кодов является более надежным способом контроля целостности, чем биты четности. В то же время, при передаче исходного сообщения P(x) возможна такая его модификация, что контрольная сумма для него и принятого искаженного сообщения P’(x) будут совпадать. Т. е. циклические коды не лишены проблемы возникновения коллизий.

Выбор и применение на практике вида порождающего полинома определяется требованиями производительности и минимизации возникновения коллизий. В следующей таблице приведены некоторые разновидности порождающих полиномов, используемые в информационных системах.

Таблица 15.7

Разновидности порождающих полиномов

Название	Порождающий полином G(x)	Нормальное представление	Применение

CRC-1	x + 1	0116	аппаратный контроль ошибок (нечетный паритетный бит)
CRC-4-ITU	x4 + x + 1	0316	
CRC-7	x7 + x3 + 1	0916	системы телекоммуникации, MMC, SD
CRC-12	x12 + x11 + x3 + x2 + x + 1	080F16	системы телекоммуникации
CRC-16-IBM	x16 + x15 + x2 + 1	800516	USB, ANSI X3.28
CRC-16-CCITT	x16 + x12 + x5 + 1	102116	X.25, Bluetooth, SD, RFID
CRC-24-Radix-64	x24 + x23 + x18 + x17 + x14 + x11 + x10 + x7 + x6 + x5 + x4 + x3 + x + 1	864CFB16	OpenPGP
CRC-32-IEEE 802.3	x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1	1EDC6F4116	V.42, MPEG-2, PNG

CRC-64-ISO	x64 + x4 + x3 + x + 1	1B16	HDLC

Нормальное представление полинома указывается в стандартах, как правило, в шестнадцатеричном виде. При этом преобразование битовой строки полинома выполняется без учета старшего единичного бита. Например, для CRC-12 битовая строка порождающего полинома выглядит 11000000011112. Отбросив старший бит (1000000011112) и преобразовав в шестнадцатеричный вид, получаем 080F16.

Другой вариант использования контрольных сумм - проверка целостности хранимых файлов с целью обнаружения их искажения (например, вирусами) или подмены (например, rootkit в Unix-системах). В этом случае обычно применяют хеш-образы файлов, которые хранятся в защищённом месте и периодически используются для контроля целостности файлов. Хеш-образы активно применяют для проверки целостности скачиваемых файлов. В частности на многих сайтах, помимо дистрибутивов ПО, выкладываются также их контрольные хеш-образы.

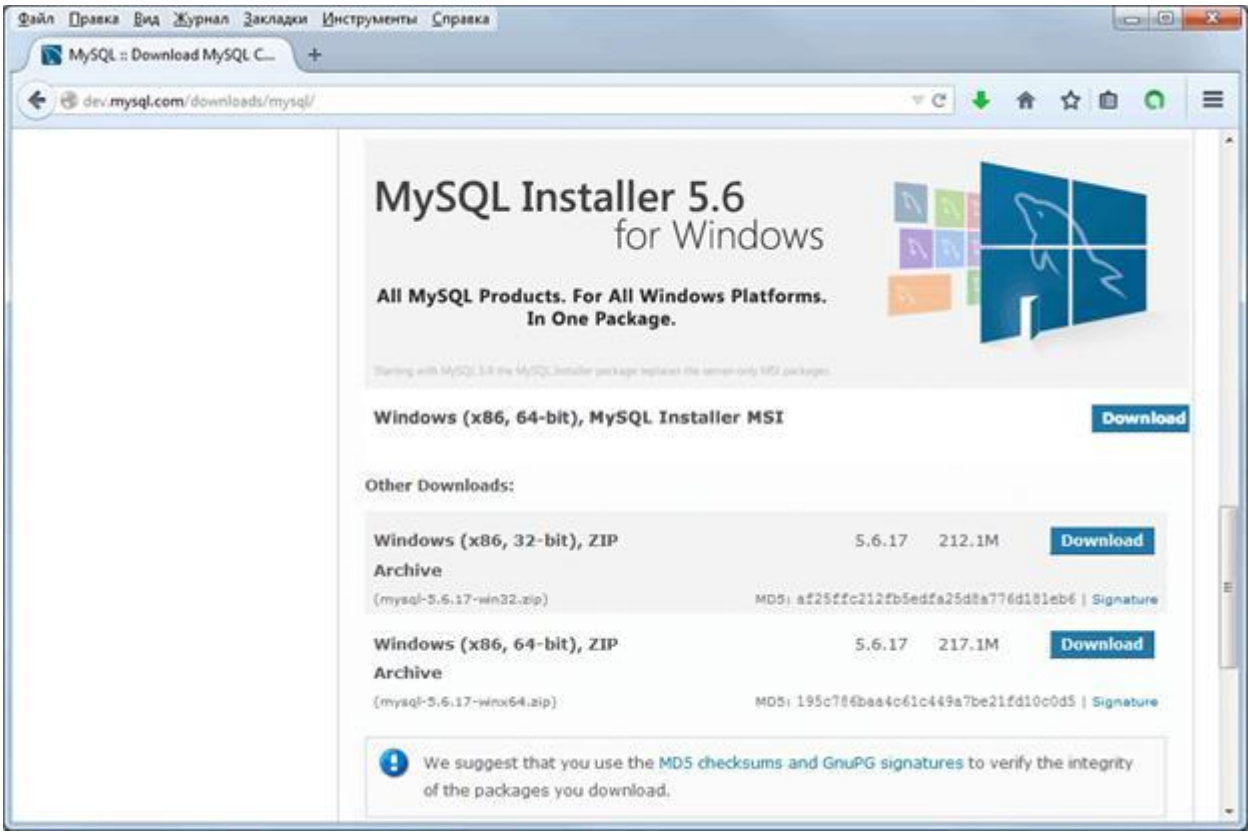

Рис.15.5. Контрольные суммы (хеш-образы MD5) для дистрибутивов СУБД MySQL

15.5. Использование ECC

Одним из существенных недостатков рассмотренных выше способов контроля целостности является невозможность устранения обнаруженных ошибок. В случае приема искаженных данных через сеть выполняется их повторная пересылка. Другим способом устранения этого недостатка является использование самокорректирующихся кодов. Коды, в которых возможно автоматическое исправление ошибок, называются **самокорректирующимися**. Одним из таких кодов является ECC.

ECC (англ. error-correcting code, код коррекции ошибок) – контрольные биты, присоединяемые к блоку хранимых или передаваемых данных, позволяющие определить факт нарушения целостности данных и (в некоторых случаях) исправить ошибку.

ECC являются разновидностью **кодов Хэмминга**, обозначаемых обычно (**m, k**), где **m** – количество битов в блоке данных (информационных и контрольных битов), **k** - количество контрольных битов. Формула классического кода Хэмминга имеет вид (2k – 1, 2k – k – 1), например: (7,4), (15,11), (31,26). Данная формула позволяет обнаруживать одиночную или двойную ошибку, а также исправлять одиночную ошибку, но при условии, что ошибок не может быть более двух. В случае возможности возникновения трех и более ошибок в данных использование классического кода Хэмминга некорректно.

Рассмотрим алгоритмы получения контрольных битов и проверки целостности данных на примере информационной битовой строки 100100101112 (m = 15 бит, k = 4 бит).

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11
Обозначение бита	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
Значение бита, X	1	0	0	1	0	0	1	0	1	1	1

Алгоритм получения контрольных бит (кодирования).

1) В блок исходных данных вставляются контрольные биты (r1 .. r4) таким образом, чтобы номера их позиций представляли собой целые степени двойки, начиная с нуля - (20, 21, 22, 23) = (1, 2, 4, 8). Первоначально контрольные биты устанавливаются равными нулю.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
Значение бита, XR	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1

2) К полученной таблице добавляются строки, число которых соответствует количеству контрольных битов. В столбцы добавленных строк записываются двоичные представления номеров позиций битов, причем порядок следования битов двоичного представления будет обратный - младший бит располагается в верхней строке, старший - в нижней.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1		
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	r_1	
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	r_2	
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	r_3	
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	r_4	

3) Определение значений контрольных битов выполняется по формуле

$$r_j = \left(\sum_{i=1}^{m} XR_i * N_{ji} \right) mod\ 2, \quad (15.8)$$

где j – номер контрольного бита, j = 1 .. k;

i – номер позиции бита, i = 1 .. m.

В терминах матричной алгебры получение вектора контрольных битов **R** является результатом перемножения вектора **XR** на транспонированную матрицу **N** (строки заменены на столбцы) по модулю 2.

R = (XR ^{x} NT) mod 2. (15.9)

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1		
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	r_1	1
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	r_2	1
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	r_3	0
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	r_4	0

Таким образом, блок данных вместе с контрольными битами будет выглядеть 10010010111 11002.

Алгоритм проверки целостности данных (декодирования).
1) Определение вектора синдромов **S**. Процедура получения вектора аналогична п.3 алгоритма кодирования. При этом в строку «Значение бита» сразу записываются значения контрольных битов. После перемножения вектора **XR'** на транспонированную матрицу **N** получается вектор синдромов **S**

S = (XR' ^{x} NT) mod 2. (15.10)

2) Определение нечетного паритетного бита **pb** для вектора **XR'**.

$$pb = \left(\sum_{i=1}^{m} XR'_i \right) mod\ 2. \quad (15.11)$$

3) Анализ вектора синдромов и паритетного бита. Возможны следующие случаи.

3.а) Ошибок нет. Вектор синдромов состоит из нулей и паритетный бит равен 0.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR'	1	1	1	0	0	0	1	0	0	0	1	0	1	1	1	pb	0
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	s_1	0
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	s_2	0
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	s_3	0
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	s_4	0

3.б) Одиночная исправимая ошибка. В векторе синдромов есть единицы и паритетный бит равен 1.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR'	1	1	1	0	0	0	0	0	0	0	1	0	1	1	1	pb	1
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	s_1	1
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	s_2	1
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	s_3	1
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	s_4	0

Переставив биты из вектора синдромов S = (1 1 1 0) в обратном порядке получают номер ошибочного бита 01112 = 710 (x4), который можно исправить, инвертировав его.

3.в) Двойная неисправимая ошибка. В векторе синдромов есть единицы и паритетный бит равен 0.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR'	1	1	1	1	0	0	0	0	0	0	1	0	1	1	1	pb	0
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	s_1	1
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	s_2	1
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	s_3	0
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	s_4	0

Обнаруженную двойную ошибку невозможно исправить (по аналогии со случаем 3.б), т. к. вектор синдромов S = (1 1 0 0) показывает, что ошибка в бите с номером 00112 = 310 (x1), хотя на самом деле ошибки в битах 4 (r3) и 7 (x4).

Как отмечалось выше, если количество ошибок более двух, использование кода Хэмминга некорректно.

Если в блоке данных нечетное количество ошибок, то паритетный бит равен 1. По этому признаку можно установить наличие ошибки в данных, но не количество этих ошибок. Например, тройная ошибка может дать различные вектора синдромов, в т. ч. и нулевой.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR'	1	1	0	1	0	0	0	0	0	0	1	0	1	1	1	pb	1
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	s_1	0
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	s_2	0
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	s_3	0
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	s_4	0

При четверной ошибке возможна ситуация, что алгоритм декодирования вообще не обнаружит факт наличия ошибки, т. к. паритетный бит будет равен 0, а вектор синдромов – нулевой.

Номер позиции бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Обозначение бита	r_1	r_2	x_1	r_3	x_2	x_3	x_4	r_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}		
Значение бита, XR'	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	pb	0
Двоичное представление номера позиции бита, N	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	s_1	0
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	s_2	0
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	s_3	0
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	s_4	0

ECC на базе кодов Хэмминга используется при хранении данных как на магнитных носителях, так и в оперативной памяти.

15.6. Использование ЭЦП

.

Предыдущий способ для передачи и хранения критичных данных (конфиденциальной информации или исполняемых компонентов (ActiveX, exe-модулей и т. д.)) не годиться. Злоумышленник может подменить передаваемые данные и контрольную сумму к ним, т. к. алгоритм генерации контрольных данных не является секретом и может быть известен любому желающему.

В случае использования ЭЦП эта проблема решается. Кроме того, можно также убедиться, что сообщение пришло от того, кто знает секретный ключ владельца подписи.

В Windows 2000 и XP все системные файлы и драйверы (с расширениями sys, dll, ocx, ttf, fon и exe) защищены с помощью ЭЦП. Данная функция работает в фоновом режиме и защищает все файлы, установленные программой Windows Setup. Если новая версия не является корректной, то она заменяется копией из каталога %SystemRoot%\System32\Dllcache или диска дистрибутивом. По

умолчанию эта функция активирована и позволяет выполнить замену файлов только в случае установки следующих видов ПО:

- сервисных пакетов Windows;

- дистрибутивные пакеты типа Hotfix;

- обновление версии ОС с помощью Winnt32.exe;

- ПО Windows Update.

Для установки или просмотра опций защиты системных файлов необходимо выбрать «Мой компьютер \ пункт меню Свойства \ вкладка Оборудование \ кнопка Подписывание драйверов».

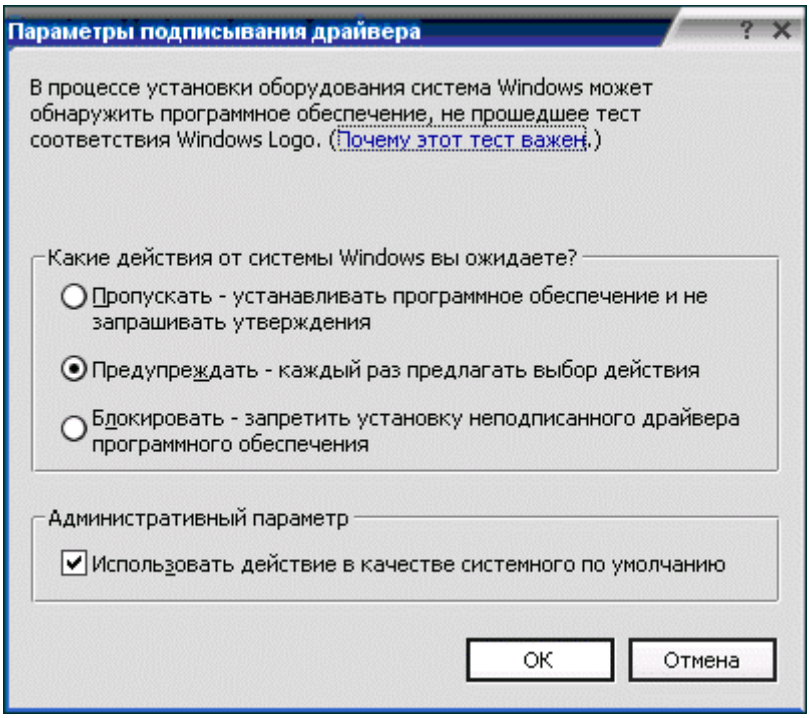

Рис.15.6. Параметры подписывания драйверов

Для проверки системных файлов можно воспользоваться стандартной утилитой командной строки Sfc. exe (System File Checker). В случае обнаружения некорректной версии файла она будет при возможности заменена.

С помощью стандартной утилиты sigverif. exe можно выявить все файлы в папке %SystemRoot% не имеющие подписи.

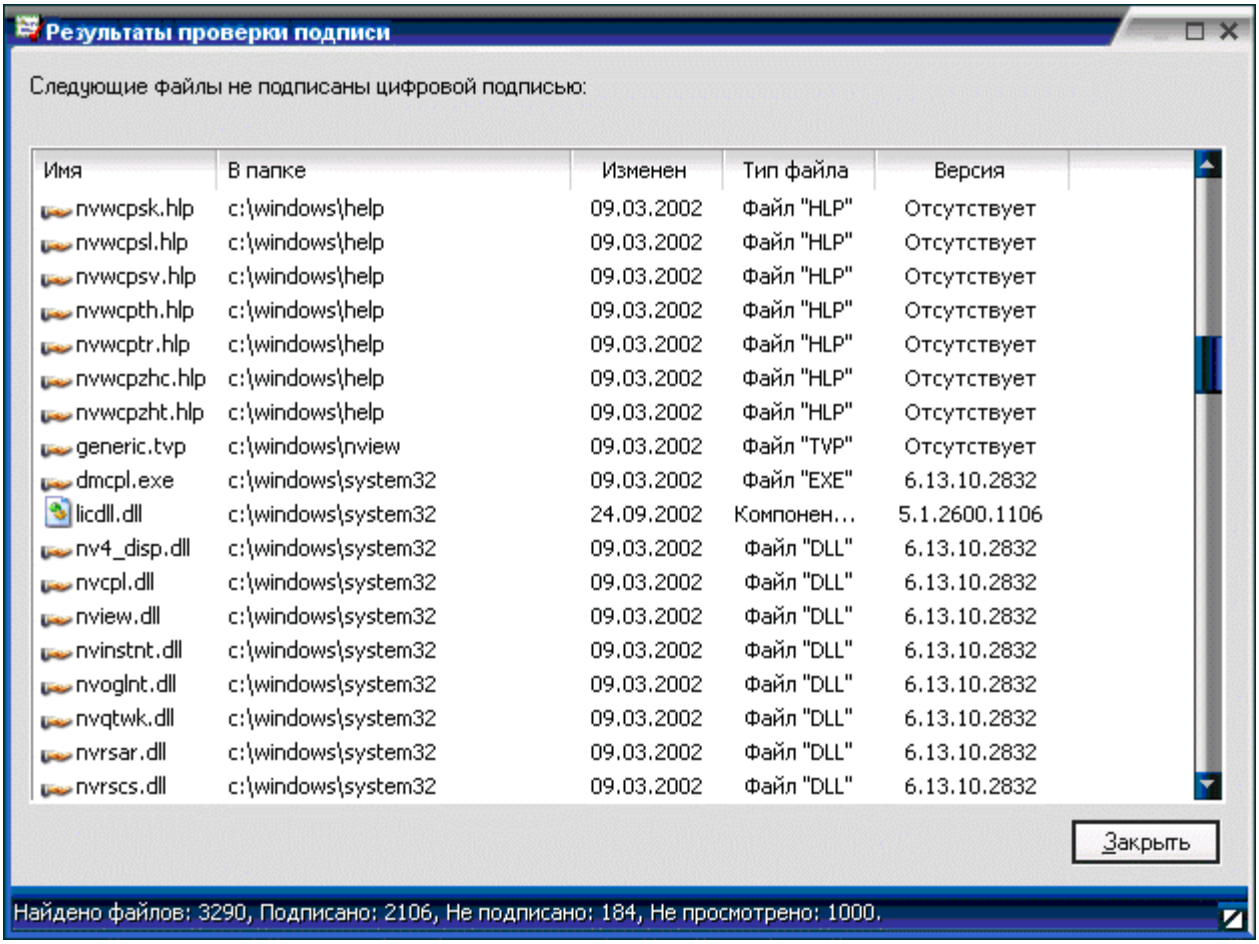

Рис.15.7. Результаты проверки подписи драйверов

15.7. Использование MAC-кодов

Код проверки подлинности сообщения (Message Authentication Code, MAC, имитовставка), известный также как код проверки подлинности данных (Data Authentication Code, DAC), представляет собой однонаправленную хеш-функцию с добавлением закрытого ключа. Значение хеш-образа зависит не только от документа, но и от ключа. В качестве примера MAC-кода можно привести последний 64-битовый блок, получаемый при шифровании DES в режиме CBC (Cipher Block Chaining, сцепление блоков шифра) [8].

Сфера применения MAC-кодов аналогична ЭЦП, только в отличие от ЭЦП для их генерации используется симметричное шифрование.

15.8. Комбинированные методы (на примере жестких магнитных дисков)

Структура сектора жесткого магнитного диска на следующем рисунке.

Рис.15.8. Структура сектора жесткого магнитного диска

Сектор состоит из заголовка-идентификатора и области данных. Начало сектора помечается специальным байтом - адресным маркером (1). Он служит для сообщения контроллеру о том, что сектор находится под головкой. Затем следуют ячейки, в которых содержится уникальный адрес сектора в формате CHS (2) и его контрольная сумма для проверки целостности записанного адреса (3). 512 байт данных пользователя помещаются в отдельном поле (4), к которому при записи добавляется несколько десятков байт избыточной информации, предназначенной для коррекции ошибок чтения с помощью ECC-кода (5). Рядом с данными размещается 4 байта циклической контрольной суммы (CRC) данных, которая служит для проверки целостности данных пользователя, и сообщения системе коррекции ошибок при ее нарушении (6). Для более надежной работы сектора при колебаниях скорости вращения имеются байты-пробелы (7). У некоторых винчестеров имеется дополнительный байт после AM - в нем сектор помечается как bad.

Bad-секторы (англ. bad - плохой, негодный) есть на любых винчестерах. Как бы тщательно не были изготовлены их диски, на каждом из них найдется несколько мест, запись или чтение которых сопровождается ошибками. Кроме того, на них встречаются участки поверхности, которые могут со временем перерасти в дефекты. Поэтому каждый накопитель после изготовления на заводе, проходит тщательное тестирование, в процессе которого выявляются испорченные секторы. Они помечаются как негодные и заносятся на диск в специальную таблицу - дефект-лист (**P-list**, Primary-list). Кроме заводского P-list'a дефектов накопитель имеет еще **G-list** (Grown-list) - в него заносятся сведенья о bad-секторах, обнаруженных в процессе эксплуатации.

Для того, чтобы у пользователя создать иллюзию отсутствия дефектов на диске, на нем имеется резервная область с запасными дорожками или запасной сектор на каждой дорожке. Специальный «транслятор» диска пропускает через себя все обращения BIOS, ОС и прикладных программ к секторам и в случае необходимости перенаправляет их в резервную область. Только в случае переполнения G-list на диске появляются «видимые» bad-сектора.

Почти все винчестеры, выпущенные после 95-го года, имеют систему оперативного наблюдения за своим состоянием - **S. M.A. R.T.** (Self Monitoring And Reporting Technology). Эта технология позволяет в любое время оценить такие важные параметры накопителя, как количество отработанных часов, число возникших в процессе чтения/записи ошибок и многое другое.

Стандарт SMART-II регламентирует выполнение внутренней диагностики. Эта функция основана на проведении серии автономных внутренних тестов и предназначена для углубленного контроля за состоянием механики накопителя, поверхности дисков и многих других параметров. После выполнения тестов, накопитель должен записывать показания в SMART-атрибуты.

Для просмотра smart-статуса жесткого диска используют программы, называемые **smart-мониторами**. Одна из них - SmartMon (http://www. kursknet. ru/~boa/delphi/SmartView. zip, с исходными текстами на Паскале).

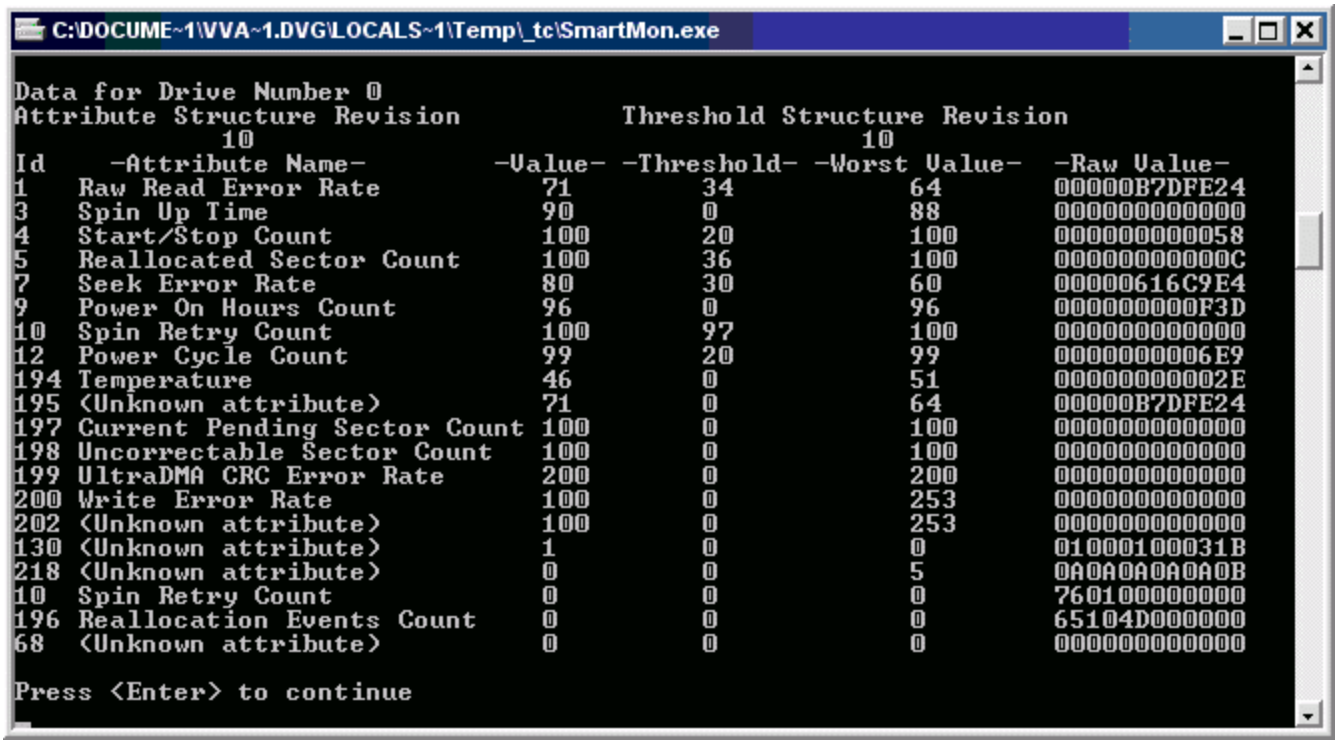

Рис.15.9. Результат проверки жесткого диска программой SmartMon

Примечание. Raw Value – фактическое значение, Threshold – допускаемое производителем пороговое значение.

Между **SMART-атрибутами** и состоянием поверхности существует некоторая взаимосвязь. Рассмотрим те из них, которые имеют прямое отношение к bad-блокам.
Raw read error rate (1): количество ошибок чтения. У многих винчестеров (например, у Seagate и Fujitsu) они всегда выше нуля, но если значение Value находится в пределах нормы, опасаться нечего. Это «мягкие» ошибки, успешно скорректированные электроникой накопителя и не приводящие к искажению данных. Опасно, когда этот параметр резко снижается за короткий срок, переходя в желтую зону. Это говорит о серьезных проблемах в накопителе, о возможном появлении bad-секторов в ближайшее время.
Reallocated sector count (5): число переназначенных секторов. Эти атрибуты показывают количество секторов, переназначенных G-list. У новых винчестеров они обязательно должны быть равны нулю. Если

их значение отличается от нуля, то это означает, что винчестер уже был в употреблении, на нем появлялись bad-сектора.
Current Pending Sector (197): этот атрибут отражает содержимое «временного» дефект-листа, присутствующего на всех современных накопителях, т. е. текущее количество нестабильных секторов. Эти секторы винчестер не смог прочесть с первого раза. Поле Value этого атрибута показывает общее количество секторов, которые накопитель в данный момент считает претендентами на переназначение в G-list. Если в дальнейшем какой-то из этих секторов будет прочитан (или переписан) успешно, то он исключается из списка претендентов. Постоянное значение этого атрибута выше нуля говорит о неполадках в накопителе.
Uncorrectable Sector (198): показывает количество секторов, ошибки в которых не удалось скорректировать ECC-кодом. Если его значение выше нуля, то это означает, что очень скоро во время записи данных ОС нарвется на этот сектор и в результате какая-нибудь важная информация или системный файл окажутся испорчены.

Посмотреть значения SMART-атрибутов можно также в программе SiSoftware Sandra.

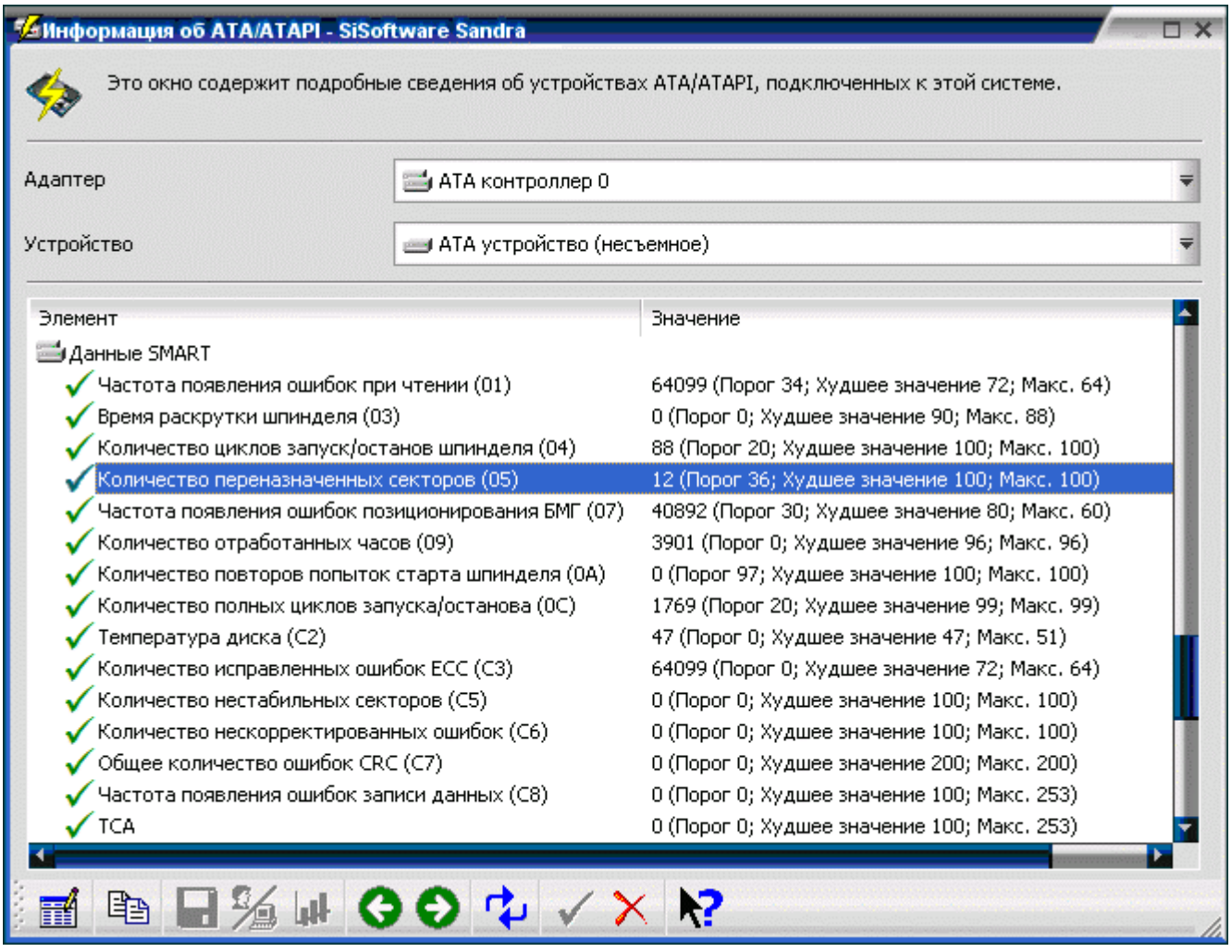

Рис.15.10. Результат проверки жесткого диска программой SiSoftware Sandra

Вопросы для самопроверки

1. Перечислите основные способы контроля целостности.

2. Что такое бит четности и как с помощью него осуществляется контроль целостности?

3. Для чего предназначена технология S. M.A. R.T.?