

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до виконання лабораторних робіт**  
**з навчальної дисципліни**  
**«Контроль та діагностика комп'ютерних систем»**

для студентів денної та заочної форми навчання  
за спеціальністю «Комп'ютерна інженерія»

2-ге видання, перероблене та доповнене

Затверджено  
редакційно-видавничою  
радою НТУ «ХПІ»,  
протокол № 2 від 27.06.2024

Харків  
НТУ «ХПІ»  
2024

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Контроль та діагностика комп'ютерних систем» для студентів денної та заочної форми навчання за спеціальністю «Комп'ютерна інженерія», 2-ге видання, перероблене та доповнене / В.І. Носков, В.І. Панченко, О.І. Баленко. – Харків: НТУ «ХПІ», 2024. – 67 с.

Укладачі: В.І. Носков, В.І. Панченко, О.І. Баленко

Рецензент доц. Н.Ю. Любченко

Кафедра комп'ютерної інженерії та програмування

## ВСТУП

Методичні вказівки містять методику виконання лабораторних робіт, метою яких є отримання студентами спеціальних навичок з діагностики та ремонту персональних комп'ютерів та ознайомлення з програмами тестування.

Сучасний персональний комп'ютер являє собою не просто складний пристрій з електронними та електронно-механічними вузлами, але й пристрій, наповнений складними операційними системами, програмними пакетами, програмами тестування і самоперевірки вузлів і блоків, які беруть участь у роботі обчислювальної машини. Персональний комп'ютер та його програмне забезпечення з часом постійно ускладнюються і з'являються нові погляди на діагностику і ремонт комп'ютерних систем і комплексів.

Раціонально складені лабораторні роботи по технічному обслуговуванню та ремонту комп'ютерних систем і комплексів дозволяють отримати спеціальні навички з діагностики та ремонту персонального комп'ютера за допомогою технічних засобів. Кожна лабораторна робота містить коротку теоретичну довідку, детальні алгоритми для проведення діагностики і опис використовуваної програми, а також практичні завдання та контрольні питання.

Розроблені лабораторні роботи призначені для студентів спеціальності 123 «Комп'ютерна інженерія», але можуть бути корисні для студентів інших спеціальностей та викладачів.

# Лабораторна робота 1

## ПРОГРАМИ ДІАГНОСТИКИ ПК

**Мета роботи:** вивчити та протестувати обладнання комп'ютера, визначити його характеристики.

### Хід роботи

#### 1. Основні теоретичні відомості

До основних складових частин персонального комп'ютера (ПК) відносять: системний блок, монітор, клавіатуру, мишу. У свою чергу, системний блок складається з материнської (системної) плати, блоку живлення, жорсткого диска, оптичного приводу та ін.

Розглянемо склад, призначення та характеристики деяких основних компонентів системного блоку.

Системна плата – це складна багатошарова друкована плата, на якій встановлюються основні компоненти ПК: центральний процесор (ЦП), оперативний запам'ятовуючий пристрій (ОЗП), постійний запам'ятовуючий пристрій (ПЗП), контролери базових інтерфейсів введення-виведення та ін. Швидкодія різних компонентів комп'ютера може мати відчутні відмінності. Для узгодження швидкодії на системній платі встановлюються спеціальні мікросхеми (чіпсети), які включають у себе контролер оперативної пам'яті (північний міст) і контролер периферійних пристроїв (південний міст).

Північний міст забезпечує обмін інформацією між процесором і оперативною пам'яттю. У процесорі використовується внутрішнє множення частоти, тому частота процесора в кілька разів більше, ніж частота системної шини. До північного мосту підключається шина взаємодії периферійних пристроїв PCI (Peripheral Component Interconnect bus), яка забезпечує обмін інформацією з контролерами периферійних пристроїв (частота контролерів менше частоти системної шини). Контролери периферійних пристроїв встановлюються в слоти розширення системної плати. Південний міст забезпечує обмін інформацією між північним мостом і портами для підключення периферійного обладнання.

Пристрої зберігання інформації підключаються до південного мосту по шині UDMA (Ultra Direct Memory Access – пряме підключення до пам'яті).

Для підключення периферійних пристроїв зазвичай використовується порт USB (Universal Serial Bus – універсальна послідовна шина), який забезпечує високошвидкісне підключення до комп'ютера відразу декількох пристроїв. Системні плати класифікуються за форм-фактором – це стандарт, який визначає розміри материнської плати для ПК, місця її кріплення до корпусу, розташування на ній інтерфейсів шин, портів введення/виведення і т.п. Прикладами позначення форм-факторів є ATX, Mini-ITX, Nano-ITX, MicroBTX.

ЦП – обчислювальний пристрій, виконавець машинних інструкцій, частина апаратного забезпечення комп'ютера, який відповідає за виконання операцій, заданих програмами. У даний час використовуються багатоядерні процесори на одному або декількох кристалах.

Для підвищення швидкодії використовують так зване кешування – це використання додаткової швидкодіючої пам'яті для зберігання копій блоків інформації з ОЗП.

Оперативна пам'ять – це робоча область для процесора комп'ютера, в якій під час роботи зберігаються програми і дані. Оперативна пам'ять часто розглядається як тимчасове сховище, тому що дані і програми в ній зберігаються тільки при включеному комп'ютері або до натискання кнопки скидання (reset).

У сучасних комп'ютерах використовуються запам'ятовуючі пристрої трьох основних типів: ROM (Read Only Memory) – постійний запам'ятовуючий пристрій, DRAM (Dynamic Random Access Memory) – динамічний пристрій з довільним порядком вибірки, SRAM (Static RAM) – статична оперативна пам'ять. У пам'яті типу ROM дані можна тільки зберігати, використовується така пам'ять тільки для читання даних. ROM також часто називається енергонезалежною пам'яттю, тому що будь-які дані, записані в неї, зберігаються при виключенні живлення. В ROM містяться команди запуску персонального комп'ютера (програмне забезпечення, яке завантажує систему).

Пам'ять DRAM використовується в більшості систем оперативної пам'яті сучасних ПК. Основна перевага пам'яті цього типу полягає в тому, що її осередки упаковані дуже щільно і на їх основі можна побудувати пам'ять великої ємності. Проблеми, пов'язані з пам'яттю цього типу, викликані тим, що вона динамічна, тобто повинна постійно регенеруватися.

Пам'ять SRAM названа так тому, що для збереження її вмісту не потрібно періодичної регенерації, але це не єдина її перевага. SRAM має більш високу швидкодію, ніж динамічна оперативна пам'ять, і може працювати на тій же частоті, що і сучасні процесори.

Жорсткий диск – незалежний комп'ютерний пристрій, є основним накопичувачем даних практично у всіх комп'ютерах. Мережева плата (адаптер) – периферійний пристрій, що дозволяє комп'ютеру взаємодіяти з іншими пристроями мережі. Звукова карта – плата, яка дозволяє працювати зі звуком на комп'ютері. У даний час звукові карти бувають як вбудованими в материнську плату, так і окремими платами розширення. Відеоадаптер – пристрій, що перетворює зображення, що знаходиться в пам'яті комп'ютера, у відеосигнал для монітора.

Існує декілька невеликих програм, які дозволяють швидко отримати характеристики процесора, материнської плати, встановленої пам'яті і відеокарти, визначити додаткове обладнання та програмну складову, виконати тест монітору та пам'яті.

### **1.1 Програма отримання основних даних про базові пристрої системи CPU-Z**

Програма CPU-Z (робоче вікно наведено на рис. 1.1) дозволяє визначити процесор, материнську плату, оперативну пам'ять та відеокарту.

Для материнської плати програма дозволяє отримати таку інформацію: виробник і модель; модель BIOS; чіпсет; графічний інтерфейс та ін. (рис. 1.2).

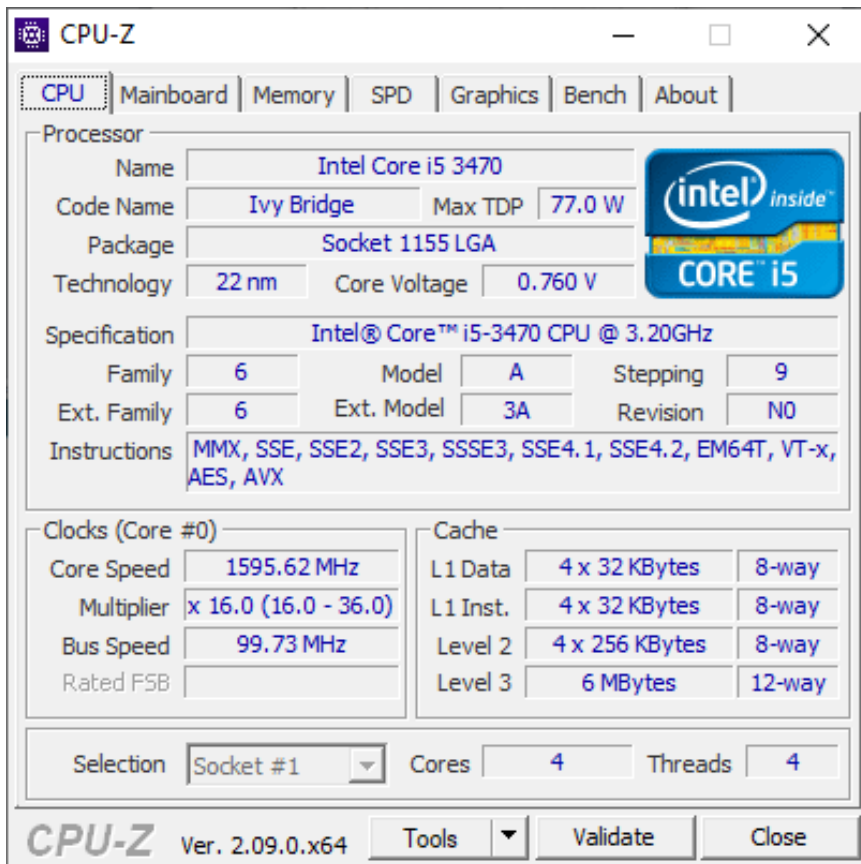


Рисунок 1.1 – Робоче вікно CPU-Z

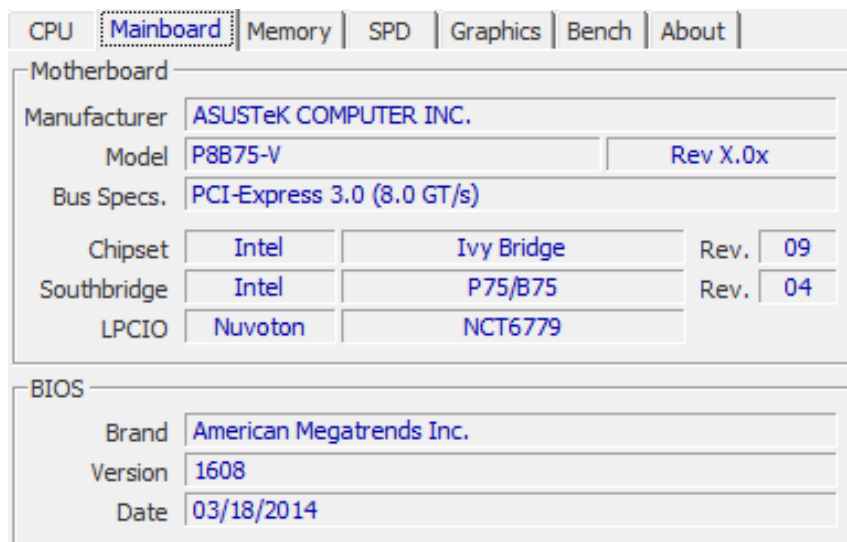


Рисунок 1.2 – Вікно з даними про материнську плату

Для пам'яті виводиться така інформація (рис. 1.3): частота; специфікації модулів; виробник; серійний номер; таблиця таймінгів.

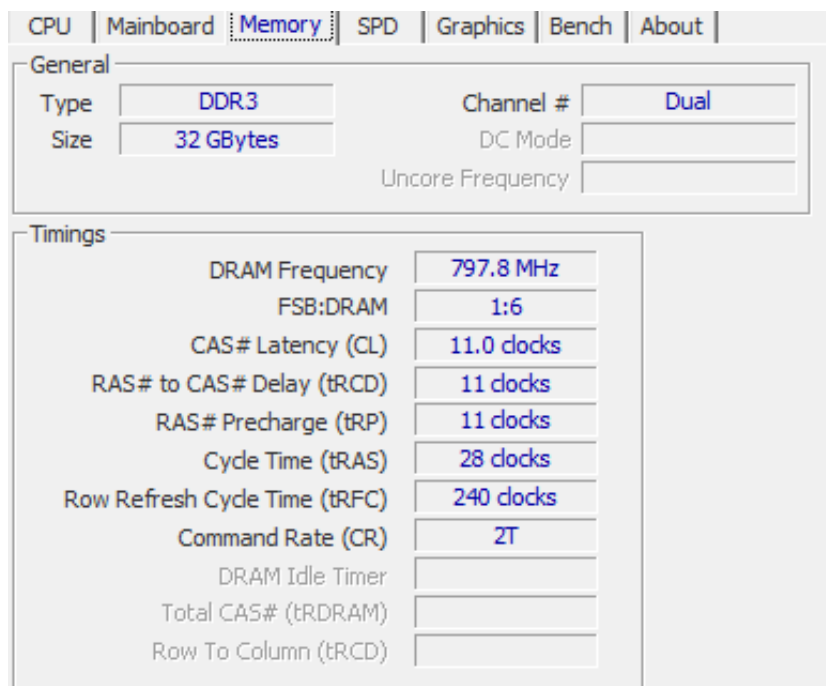


Рисунок 1.3 – Вікно з даними про пам'ять

Для графічної підсистеми виводиться така інформація: назва відеокарти; підтримка DirectX; підтримка піксельного рейдера; тип пам'яті; пропускну здатність пам'яті; тип і ширина шини; частота пам'яті та ін. (рис. 1.4).

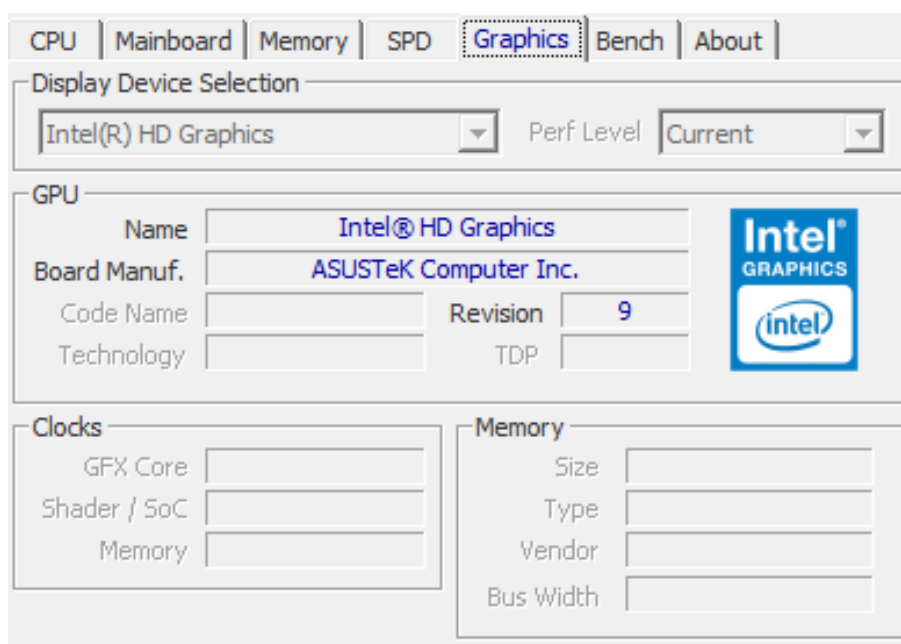


Рисунок 1.4 – Вікно з даними про графічну систему

В розділі тестування «Bench» можна порівняти характеристики продуктивності встановленого процесора з обраним еталонним процесором (рис. 1.5).

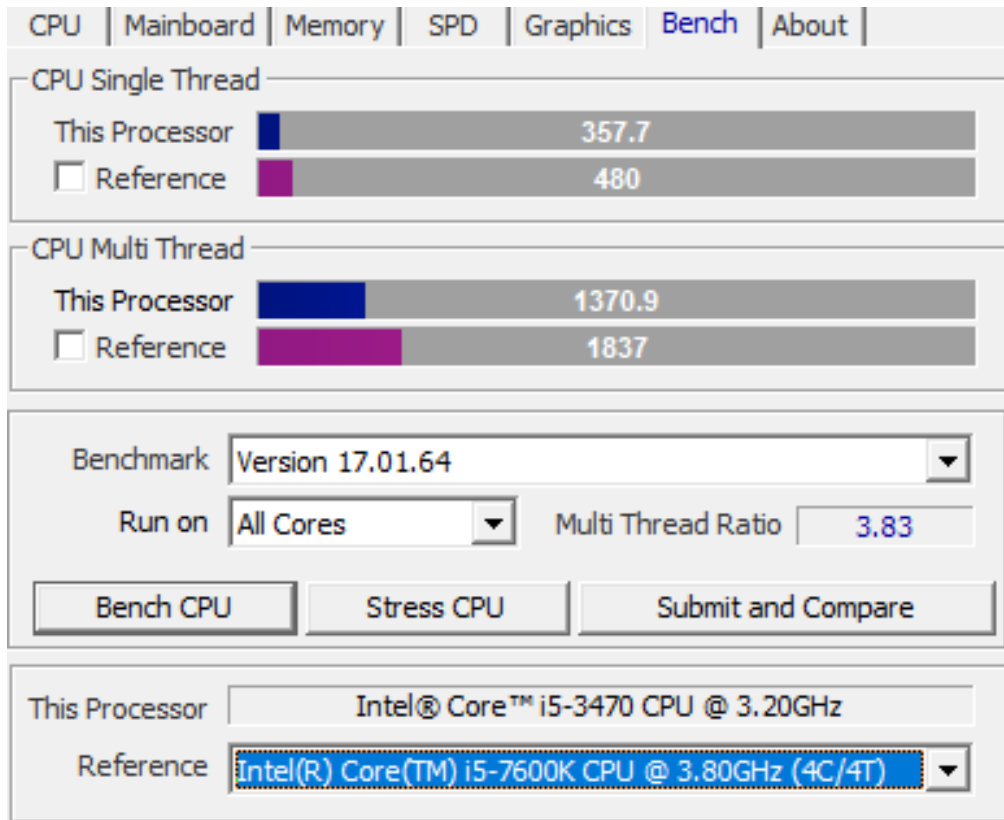


Рисунок 1.5 – Вікно тестування процесора

## 1.2 Програма моніторингу апаратного забезпечення HWMONITOR

Програма HWMonitor дозволяє виконувати моніторинг апаратного забезпечення, збирати і відображати дані з датчиків: напруги, температури і швидкості вентиляторів. Програма може визначати не тільки дані з материнської плати і процесора, але також інформацію про жорсткі диски (при наявності підтримки S.M.A.R.T.) і відеокарти.

В головному вікні програми (рис. 1.6) виводиться список, в якому представлені різні апаратні складові ПК. У програмі відображається напруга на материнській платі, показання температурних датчиків на ній, швидкість обертання вентиляторів. Датчики процесора повідомляють програмі температуру

на кожному з його ядер, а також енергоспоживання. У програмі відображається і відеокарта, датчик на ній повідомляє про температуру, таку ж інформацію надає і датчик жорсткого диска. Після інсталяції програми на портативний пристрій, наприклад, ноутбук, можна побачити відомості про батарею.

Sensor	Value	Min	Max
Temperatures			
Fans			
SYSFANIN	1829 RPM	1819 RPM	1851 RPM
CPUFANIN	1110 RPM	1087 RPM	1148 RPM
AUXFANIN1	2002 RPM	2002 RPM	2027 RPM
Utilization			
Intel Core i5 3470			
Voltages			
VID (Max)	1.026 V	0.831 V	1.071 V
Temperatures			
Package	40.0 °C	33.0 °C	43.0 °C
Cores (Max)	36.0 °C	34.0 °C	42.0 °C
Powers			
Package	30.24 W	19.55 W	36.15 W
IA Cores	11.82 W	1.17 W	16.15 W
GT	0.01 W	0.00 W	0.81 W
Uncore	18.41 W	18.28 W	19.38 W
Utilization			
Processor	14.1 %	2.6 %	37.0 %
Clocks			
Core #0	1596 MHz	100 MHz	3590 MHz
Core #1	1596 MHz	100 MHz	3590 MHz
Core #2	1596 MHz	100 MHz	3590 MHz
Core #3	3191 MHz	100 MHz	3590 MHz

Рисунок 1.6 – Робоче вікно HWMonitor

### 1.3 Програма тестування монітору PASSMARK MONITORTTEST

PassMark MonitorTest – це інструмент, який дозволяє користувачам досліджувати якість та продуктивність монітора. Програмне забезпечення MonitorTest створює тестові шаблони на екрані з різною роздільною здатністю та глибиною кольорів для перевірки оптимальної візуальної продуктивності. Кожен з тестових екранів був створений для тестування певного аспекту продуктивності моніторів (рис. 1.7).

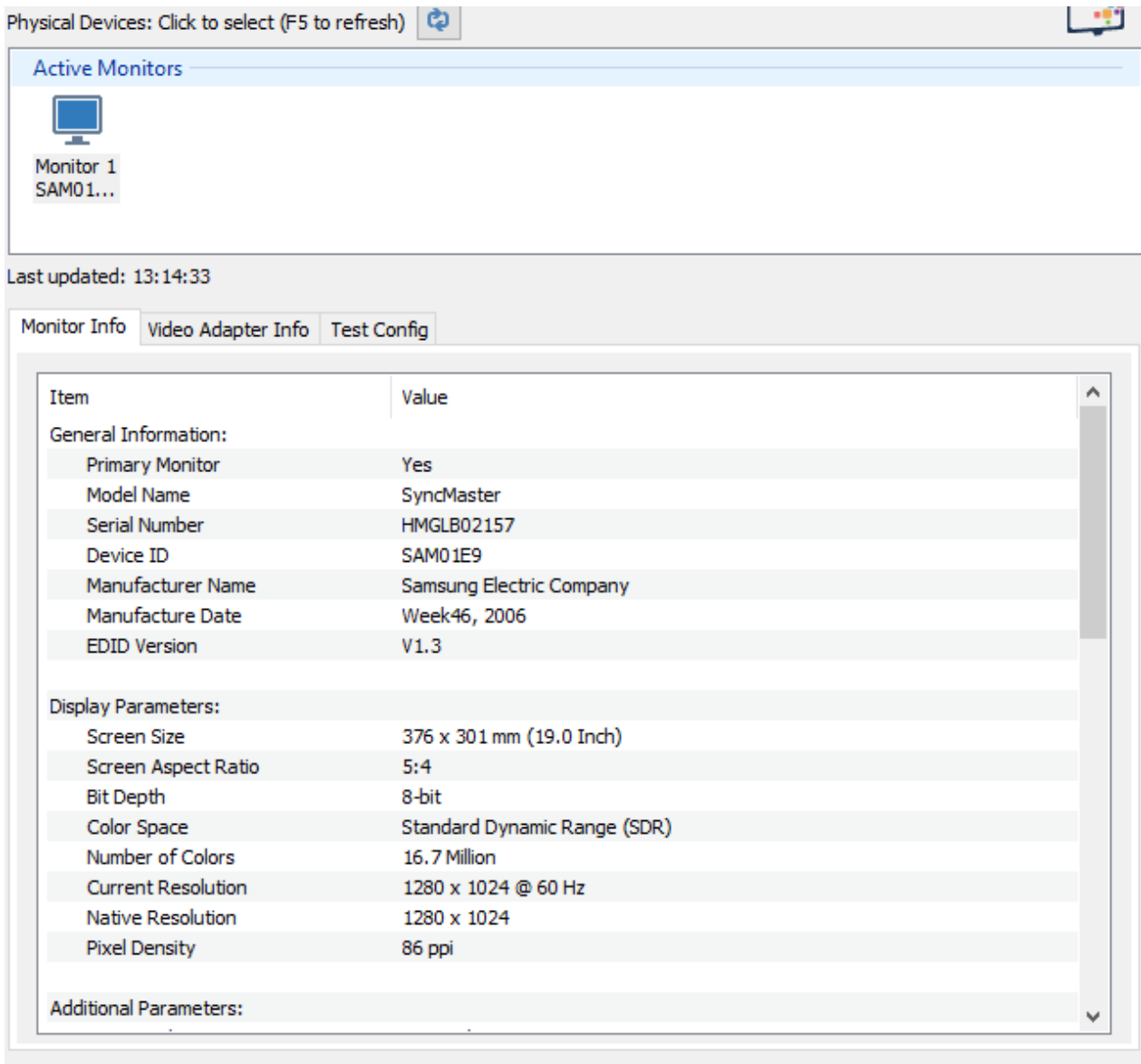


Рисунок 1.7 – Робоче вікно PassMark MonitorTest

Програма включає наступні тести: майстер-екран (рис. 1.8); контрастність; шрифти; вимірювання гама; тестування сенсорного екрана та ін.

В програмі реалізована: підтримка систем Multi-Monitor, включаючи тестування всіх моніторів одночасно; відображення інформації системи монітора та відеоадаптера; підтримка всіх доступних роздільних здатностей і глибини кольорів; підтримка всіх типів дисплеїв; підтримка різних відеорежимів.

Якщо на екрані присутні спотворення у вигляді «подушки», «трапеції» або спостерігаються спотворення пропорцій будь-якого з кіл, то слід налаштувати монітор. Налаштування виконується за допомогою кнопок управління

налаштуванням монітора, розташованих на самому моніторі, шляхом регулювання розміру по вертикалі і горизонталі, а також інших параметрів. Якщо за допомогою регулювань не вдається встановити правильні пропорції, то це ознака апаратної несправності.

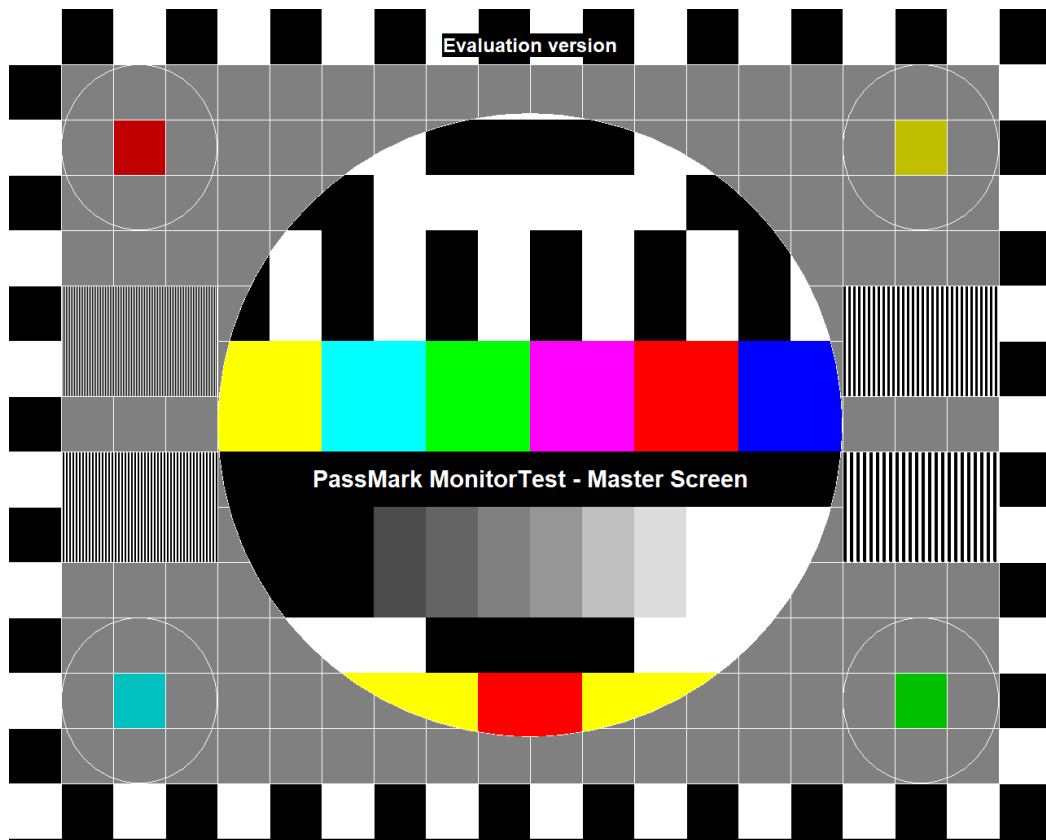


Рисунок 1.8 – Майстер-екран PassMark MonitorTest

Якщо зображення на екрані нечітке (спостерігається розмитість ліній візерунка), то причиною може бути порушення фокусування або неправильне зведення променів. Деякі монітори містять пункт меню, що дозволяє усунути цю проблему, але в більшості випадків це є свідченням апаратної несправності монітора.

У РК-дисплеях бувають «биті» піксели. Цей дефект полягає в наявності білих точок, які постійно світяться, або постійно погашених (не світяться) чорних точок. Подібна несправність не зникає, оскільки є результатом згорілих транзисторів матриці монітора. Якщо, наприклад, на зеленому кольорі ви

спостерігаєте чорну точку, якої немає на всіх інших кольорах, значить ви виявили зелений «битий» піксель. А якщо на чорному кольорі присутня кольорова або біла точка, то це ознака того, що одна з кольорових точок або всі три (червона, зелена і синя) не вимикаються і такий піксель буде світитися завжди.

#### **1.4 Програма тестування пам'яті MEMTEST86 (MEMTEST86+)**

Програма Memtest записує в кожен блок пам'яті інформацію, а потім зчитує її і перевіряє на помилки. У процесі тестування утиліта робить кілька проходів, що дозволяє виявити і скласти список bad-блоків пам'яті в форматі BadRAM. Ця програма запускається за допомогою власного завантажувача, тому наявність операційної системи для її роботи не обов'язкова.

Внаслідок того, що пам'ять перевіряється без операційної системи, то потрібно програму встановити на зовнішній носій та виконати завантаження ПК з цього зовнішнього пристрою. Одразу після завантаження програми починається тест пам'яті. Повний тест займає кілька годин для кожної планки пам'яті. У програмі MemTest86+ реалізовано 9 тестів, які виконуються циклічно:

- Test 0 [Address test, walking ones, no cache] – для визначення проблем з адресацією пам'яті;
- Test 1 [Address test, own address] – більш поглиблений тест для визначення проблем з адресацією пам'яті;
- Test 2 [Moving inversions, ones & zeros] – швидка перевірка на апаратні або важко вловимі помилки;
- Test 3 [Moving inversions, 8 bit pat] – те ж саме, тільки використовується 8-бітний алгоритм проходження нулів і одиниць;
- Test 4 [Moving inversions, random pattern] – для виявлення проблем з data sensitive;
- Test 5 [Block move, 64 moves] – для пошуку проблем у схемах пам'яті;
- Test 6 [Moving inversions, 32 bit pat] – ефективний для визначення data sensitive errors;
- Test 7 [Random number sequence] – перевіряє помилки запису пам'яті;

- Test 8 [Modulo 20, ones & zeros] – для визначення прихованих помилок за допомогою кеша і буферизації, які не виявили попередні тести;
- Test 9 [Bit fade test, 90 min, 2 patterns] – особливий тест, який можна запустити вручну. Запам'ятовує адреси в пам'яті, після чого «засинає» на півтори години. Після цього перевіряє чи не змінилися біти в адресах.

```

PassMark MemTest86 V10.5 AMD Ryzen 9 5900X 12-Core
Clk/Temp : 3701 MHz / 42C | Pass 21% #####
L1 Cache : 64K 258.2 GB/s | Test 49% #####
L2 Cache : 512K 106.2 GB/s | Test 3 [Moving inversions, ones & zeroes]
L3 Cache : 65536K 46.9 GB/s | Address : 0x100000000 - 0x82F300000
Memory : 31.9G 42.7 GB/s | Pattern : 0x00000000 RAM Temp : 31C
RAM Config: DDR4 3600MT/s / x2 Channel / G Skill Int1 F4-3600C16-16GT2NC
-----
CPU: 0123456789ABCDEF GHIJKLMN | CPUs Found: 24
State: |DWDWDWDWDWDWD\N\D\D\D\D | CPUs Started: 12 CPUs Active: 12
-----
Time: 0:00:44 AddrMode: 64-bit Pass: 1 / 4 Errors: 18

[ECC Errors detected] Test: 4 (Ch,Sl,Rk,Bk,Rw,C1) : (1,1,FFFF,FFFF,1FF,1FF)
[ECC Errors detected] Test: 4 Channel/Slot: N/A
Test: 5 Addr: 3D65D70A Expected: 00000000FFFFFFFF Actual: 0000000000000000
Test: 5 Addr: 3D65D712 Expected: 00000000FFFFFFFF Actual: 0000000000000000
Test: 5 Addr: 3D65D71A Expected: 00000000FFFFFFFF Actual: 0000000000000000
[ECC Errors detected] Test: 5 Addr: 3D65D712
[ECC Errors detected] Test: 5 Addr: 3D65D712
[ECC Errors detected] Test: 5 (Ch,Sl,Rk,Bk,Rw,C1) : (1,1,FFFF,FFFF,1FF,1FF)
>[ECC Errors detected] Test: 5 Channel/Slot: N/A
[ECC Errors detected] Test: 4 Addr: 3D65D712
[ECC Errors detected] Test: 4 Addr: 3D65D712
(ESC) / (c) onfiguration

```

Рисунок 1.9 – Вікно програми MemTest86

## 2. Індивідуальні завдання

1. Запустити програми CPU-Z, HWMonitor.
2. Отримати дані про систему, материнську плату, процесор, пам'ять, відеокарту.
3. Порівняти результати, отримані з різних програм. Визначити, яка з програм надає найбільш повну та точну інформацію про складові частини ПК.
4. Запустити програми PassMark MonitorTest, MemTest86.
5. Зробити висновки щодо результатів тесту монітора та пам'яті. У разі

помилки в оперативній пам'яті виконати підготовку та запуск тесту програми MemTest86 для детальної перевірки пам'яті.

### **3. Зміст звіту**

1. Тема та мета роботи.
2. Результати виконання завдань.
3. Висновки.

### **Контрольні запитання**

1. Які основні характеристики дозволяє визначити утиліта CPU-Z?
2. Назвіть типи спотворень зображення на моніторі, способи їх визначення та усунення.
3. Яким чином можна визначити наявність «битих» пікселів на моніторі?
4. Наведіть алгоритм тестування оперативної пам'яті при наявності декількох модулів ОЗП.

## Лабораторна робота 2

### ПРОГРАМИ ТЕСТУВАННЯ ЖОРСТКИХ ДИСКІВ

**Мета роботи:** виконати тестування жорсткого диску.

#### Хід роботи

#### 1. Основні теоретичні відомості

##### 1.1 Програма тестування жорстких дисків HDDExpert

Жорсткий диск є складним і високотехнологічним пристроєм, що складається з обертових магнітних дисків, головок читання і запису, двигуна приводу дисків і схеми управління. Для забезпечення його коректної роботи використовуються системи позиціонування магнітних головок, стабілізації швидкості обертання магнітних дисків, управління струмом запису даних та ін. Управління всіма системами виконує мікроконтролер, розташований на платі управління. Мікроконтролер також виконує самодіагностику, виправляє виникаючі помилки і забезпечує доступ до даних, що зберігаються на диску.

До жорсткого диска пред'являються високі вимоги щодо його надійності, так як від його працездатності залежать не тільки працездатність операційної системи і додатків, але і збереження даних на ньому. Всі сучасні жорсткі диски забезпечені технологією S.M.A.R.T. (Self Monitoring, Analysis and Reporting Technology – технологія самодіагностики, аналізу та звітності). Ця технологія призначена для підвищення надійності та безпеки даних на жорсткому диску. Суть самодіагностики полягає в тому, що сам жорсткий диск відстежує стан своєї працездатності і здатний заздалегідь попередити користувача про свій передаварійний стан. Це дозволить користувачеві своєчасно зробити резервну копію даних і підготуватися до заміни накопичувача.

HDDExpert – безкоштовна утиліта, принцип роботи якої полягає в перетворенні SMART-інформації накопичувачів у зручні для сприйняття та аналізу дані. Програма надає повну інформацію про жорсткі диски комп'ютера (продуктивність, помилки, попередження, температура) і має можливість тестування їх продуктивності.

Програма зчитує показники S.M.A.R.T. та переводить їх у зручні для сприйняття та аналізу візуальні дані. Вся інформація класифікується за напрямками тестування – продуктивність, температура, помилки, попередження тощо. Залежно від знайдених проблем та несправностей утиліта дає рекомендації щодо їх усунення та виправлення. Головне вікно програми наведено на рис. 2.1.

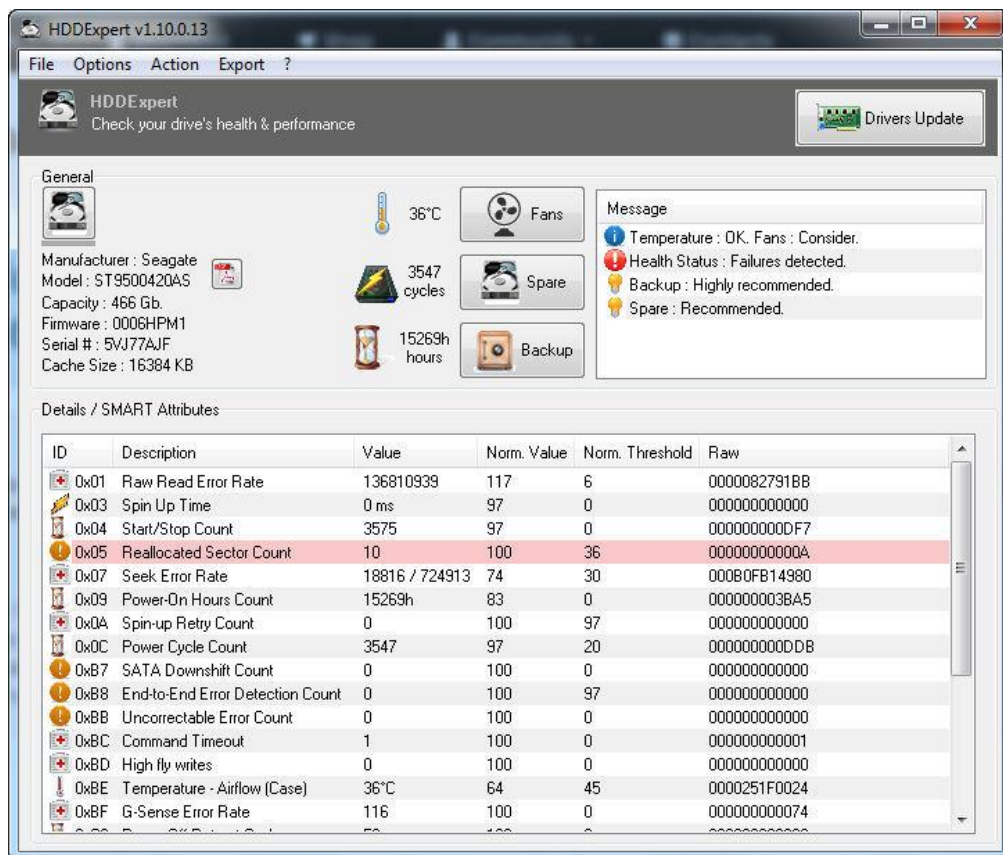


Рисунок 2.1 – Головне вікно програми HDDExpert

Основні можливості програми:

- очищення атрибутів декодування S.M.A.R.T.;
- класифікація атрибутів S.M.A.R.T.: помилки, продуктивність, температура та ін.;
- рекомендації щодо обслуговування: вентилятори, оновлення, резервне копіювання в залежності від несправності.

Стан працездатності жорсткого диска оцінюється по кількох параметрах його роботи, які називаються атрибутами надійності (attributes). Кожен атрибут

має свій номер – ID (ідентифікатор) і оцінюється за своїм кількісним значенням (value). Значення можуть змінюватися від 0 до 100 (для окремих параметрів до 255). Чим вище поточне значення атрибута надійності, тим менша ймовірність відмови по контрольованому ним параметру. Виробники жорстких дисків визначають мінімальне значення атрибута надійності, при якому гарантується його нормальна робота (Threshold – межа, поріг). При цьому атрибути діляться на критично важливі і некритично важливі. Досягнення некритично важливих атрибутів значення Threshold означає погіршення працездатності жорсткого диска, наприклад зменшення швидкодії. Але якщо значення критично важливих атрибутів стає нижче значення Threshold, то це фактично означає вихід з ладу всього накопичувача.

До стандартних атрибутів відносяться:

«Raw read error rate» – кількість помилок зчитування секторів з пластин;

«Throughput Performance» – рівень продуктивності диска;

«Spin up time» – час, за який пластини доходять від нуля до стандартної швидкості (вимірюється в мілісекундах);

«Number of spin-up times» – число циклів обертання і зупинки пластин, визначає ресурс диска, який обмежений певною кількістю даних циклів;

«Reallocated sector count» – якщо система виявляє помилку будь-якого з процесів, замість bad-сектора він вибирає гарний з доступного переліку резервних секторів. Атрибут демонструє, скільки є перепризначених секторів (в ідеалі їх не повинно бути взагалі), а для SSDRAW атрибут дозволяє бачити актуальне число неробочих блоків флеш-пам'яті;

«Read Channel Margin» – запас джерела читання. У сучасних моделях не відображається і не застосовується;

«Seek error rate» – кількість помилок позиціонування магнітних головок (МГ);

«Seek Time Performance» – середній показник швидкості позиціонування пристрою МГ на потрібний сектор (неактуальний для SSD);

«Power on time» – планований життєвий цикл диска. Розраховується,

виходячи з періоду, проведеного накопичувачем в активному стані. Спочатку число дорівнює 100, а в процесі вичерпання ресурсу знижується. Зі зменшенням показника псується, відповідно, і механічна частина диска;

«Spin up retries» – кількість спроб розкручування пластини, якщо це не вдалося зробити з першої спроби. Відраховується з моменту старту (в SSD не застосовується);

«Start \ stop count» – планований термін «життя». Підраховується на основі числа стартів / зупинок пластин. Схожий з атрибутом «Power on time» – так само падає з 100 до 0, RAW відображає кількість запусків / виключень;

«Soft Read Error Rate» – окремі бренди по-різному трактують цей атрибут. В одному варіанті він означає число помилок на невідновлених ECC, в іншому – навпаки, на відновлених;

«Erase / Program Cycles» – сума всіх циклів читання / запису для флеш-пам'яті за весь час експлуатації. У твердотільних пристроях параметр обмежений конкретним числом, що залежать від різновиду і виробника чіпів пам'яті;

«Translation Table Rebuild» – число подій зміни внутрішньої таблиці адрес блоків, при її помилках і подальшому поверненні параметрів. Цифра RAW демонструє кількість таких процесів на поточний момент;

«Reserved Block Count» – характеризує стан переліку запасних блоків в SSD, співвідношення видає відсоток доступних блоків, через RAW може відобразитися кількість вже застосованих резервних ділянок;

«Program Fail Count» – кількість невдалих спроб запису блоку флеш-пам'яті;

«Erase Fail Count» – параметр, аналогічний попередньому, тільки для стирання блоків;

«Wear Leveller Worst Case Erase Count» – найбільше доступне число операцій видалення щодо блоку флеш-пам'яті;

«Used Reserved Block Count» – характеризує стан пулу запасних блоків диска, відображає у відсотках число доступних блоків. RAW може демонструвати, скільки всього запасних блоків було використано;

«Unused Reserved Block Count» – параметр, аналогічний попередньому,

тільки через RAW відображено кількість блоків, що були застосовані;

«SATA Downshifts» – кількість випадків, коли було необхідно зниження швидкості передачі інформації по SATA (в 2 або 4 рази) для повноцінної передачі відомостей;

«End-to-End error» – число помилок, виявлених у буфері диска. Іноді свідчить, що RAM-буфер диска підлягає заміні;

«Reported UNC error» – число не виправлених помилок читання;

«Command timeout» – число команд, що не здійснено диском через таймаут очікування;

«High Fly writes» – кількість помилок збереження файлів, причиною яких стала некоректна висота знаходження МГ над робочою поверхнею;

«Airflow temperature» – показник температури всередині герметичного блоку пристрою;

«G-Sense Errors» – кількість аварійних збоїв роботи внаслідок механічних впливів;

«Power-off retract cycles» – число різких провалів живлення, у ситуаціях, коли воно зникало раніше фактичної команди на деактивацію накопичувача (при таких раптових збоях ресурс HDD зменшується в порівнянні зі звичайним циклом роботи – у SSD у таких ситуаціях є ймовірність зникнення таблиці внутрішнього стану);

«Load / unload cycles» – число, яке показує, скільки разів блок МГ переміщався між ділянкою паркування і площею даних. Показник падає від 100 до 0, RAW відображає поточне число переміщень;

«Hda temperature» – показник температури блоку МГ;

«Hardware ecc recovered» – кількість виправлених помилок читання;

«Reallocation events» – кількість перепризначення секторів, як при стандартному циклі роботи, так і в офлайн-режимі;

«Current pending sectors» – загальна цифра проблемних секторів, які чекає нова перевірка і, можливо, призначення іншого статусу;

«Offline scan unc sectors» – кількість bad-секторів, які знайшов пристрій при самоперевірці;

«Ultra dma crc errors» – число помилок при передачі відомостей від диска до материнської плати і навпаки;

«Write error rate» – частота прояву дефектів при записі;

«Data address mark errors» – число помилок при пошуку необхідної ділянки;

«Run out cancel» – число помилок, причиною яких стала неправильна контрольна сума при спробі усунення дефекту;

«Soft ecc corrections» – число помилок, виправлених відповідним коригуючим кодом;

«Flying height» – відхилення фактичної висоти знаходження МГ над поверхнею диска від нормального показника; якщо МГ знаходиться надто низько, може пошкодитись поверхня, якщо висота завищена – збільшується кількість помилок зчитування;

«Spin high current» – показник струму, достатнього для розкручування пластин;

«Offline seek performance» – продуктивність пошукового алгоритму при офлайн-перевірці;

«Disk shift» – відстань, на яку був відхилений пакет пластин від необхідного положення (причиною відхилення найчастіше є перегрів або дефект механічної частини);

«Torque amplification count» – демонструє, скільки разів потрібно було подати збільшений струм, щоб привести в дію пластини;

«Gmr head amplitude» – амплітуда коливань МГ;

«Media wearout indicator» – доступний memory-ресурс пристрою;

«Head flying hours» – час, на протязі якого МГ перебували в ділянці даних користувача;

«Total lbas written» – число 512-байтних блоків, збережених за весь минулий термін роботи пристрою;

«Total lbas read» – параметр, аналогічний попередньому, для прочитаних блоків;

«Read error retry rate» – кількість повторних дій читання сектора. Якщо

показник атрибута зростає, значить, має місце дефект поверхні, або неправильне функціонування голівки, що зчитує.

Актуалізація RAW-значень у параметрах атрибутів відбувається по-різному. Підсумком перевірки може бути рішення: стан накопичувача добрий або поганий. А тому, об'єктивний висновок про стан диска вдасться зробити лише після його розширеної перевірки.

## 1.2 Програма тестування жорстких дисків HD Tune

HD Tune – програма тестування жорсткого диска для Windows, яка може перевірити загальний стан жорсткого диска, виконати сканування на наявність помилок та виконати тест читання еталонного тесту.

HD Tune працює з будь-яким внутрішнім або зовнішнім жорстким диском, SSD або картою пам'яті. Головне вікно програми наведено на рис. 2.2.

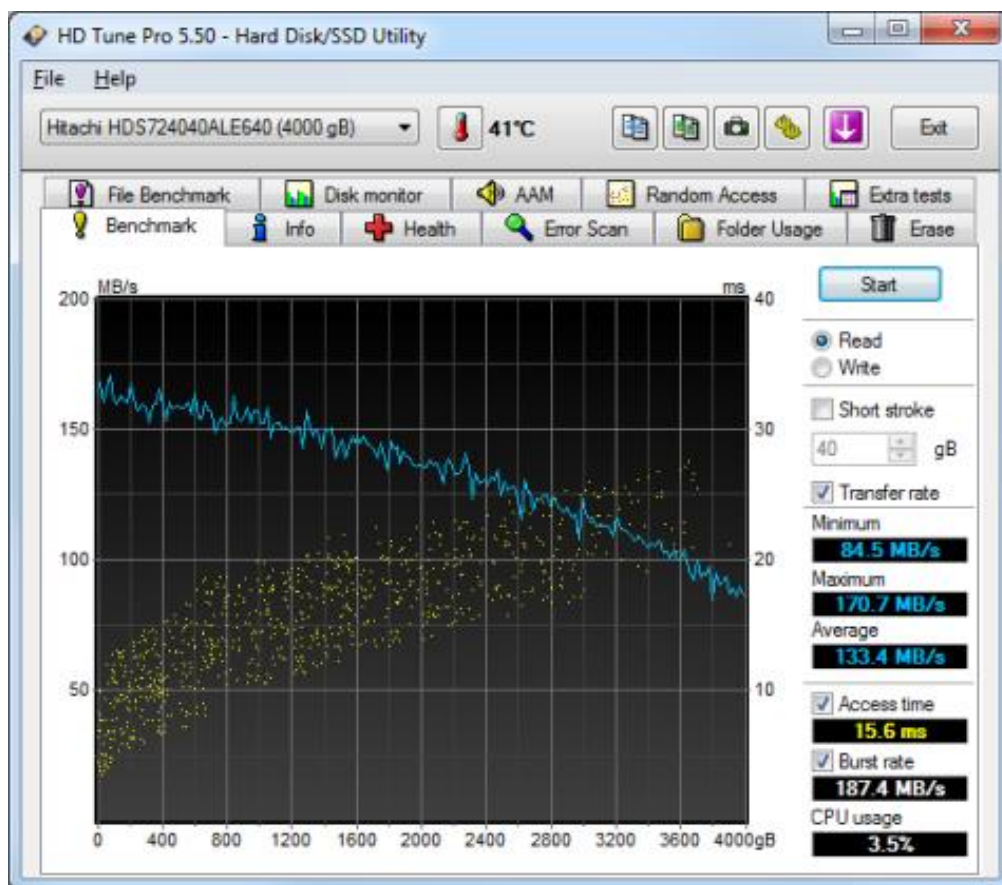


Рисунок 2.2 – Головне вікно програми HD Tune

Основними вкладками програми є Benchmark, Info, Health (SMART) та Error Scan. Налаштування еталонного тесту можна змінити на сторінці параметрів, щоб змінити швидкість тесту та розмір блоку, який використовується для читання даних із накопичувача. Коли тест запущено, видається мінімальна, максимальна та середня швидкість передачі, а також час доступу, швидкість пакетної передачі та завантаження ЦП, використані під час тесту.

Для вимірювання продуктивності використовуються два основних поняття:

- Input-Output Operations Per Second (IOPS) – кількість операцій введення-виведення в секунду, які здатний обробити пристрій. Цей параметр характеризує швидкість обробки коротких запитів, що притаманно базам даних;

- Traffic Rate – швидкість передачі даних, що вимірюється в Мб/с, що проходять по інтерфейсу введення-виведення в секунду.

HD Tune не має можливості безпосередньо вимірювати IOPS, для цього вона використовує інший показник – access time, яке вимірюється в мілісекундах. Формула перекладу виглядає таким чином:  $IOPS=1000/access\ time$ .

Перед початком тесту на продуктивність варто переконатися, що під час його виконання інші служби не звертатимуться до жорсткого диска, або ці звернення матимуть одиничний характер. Слід переконатись, що в операційній системі відсутні віруси. Антивірус та інші застосунки, які конкурують за доступ до служби дисків, повинні бути припинені. HD Tune дозволяє будувати детальний графік продуктивності.

Елементи управління головного вікна:

– Select Drive: містить всі підтримувані накопичувачі в системі. Виводиться модель накопичувача і серійний номер;

– кнопка S.M.A.R.T.: дозволяє отримати звіт про стан диску;

– кнопка TESTS: показує спливаюче меню з вибором тестів читання і запису;

– кнопка TOOLS: показує спливаюче меню для вибору доступних елементів керування та функції диску;

– кнопка More: показує спливаюче меню з елементами управління програмою.

При натисканні кнопки TESTS, спливаюче меню пропонує один з тестів. Може бути запущений тільки один тест поверхні в один час. Вікно тестів програми наведено на рис. 2.3.

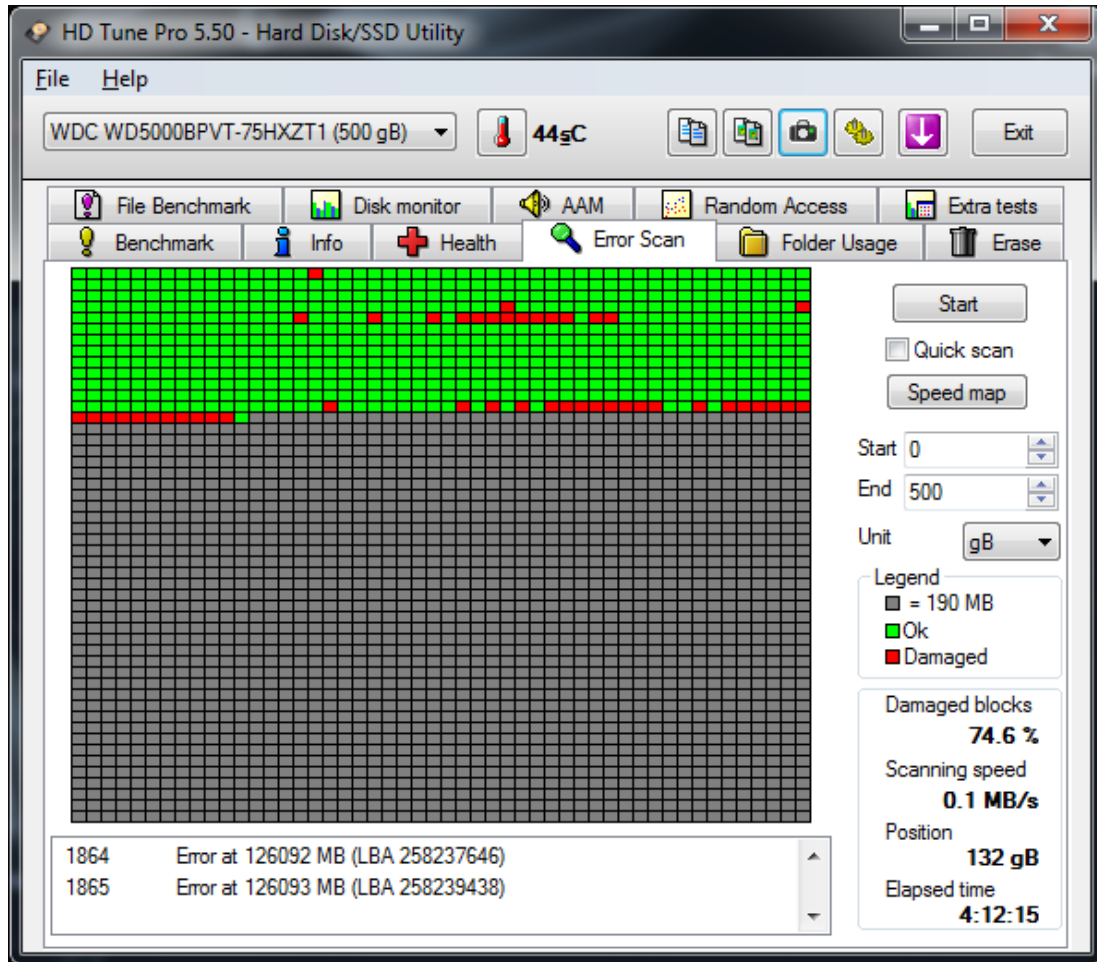


Рисунок 2.3 – Вікно тестів програми HD Tune

## 2. Індивідуальні завдання

1. Отримати перелік підключених дисків у системі та їх параметри. Занести в звіт дані про диски.
2. Виконати зчитування параметрів S.M.A.R.T. та зробити висновки щодо стану жорсткого диску.
3. Виконати тестування поверхні жорсткого диску в режимі READ.
4. Зробити висновки щодо результатів тесту, вказати кількість виявлених помилок.

### **3. Зміст звіту**

1. Тема та мета роботи.
2. Результати виконання завдань.
3. Висновки.

### **Контрольні запитання**

1. Що таке S.M.A.R.T.?
2. Які з параметрів S.M.A.R.T. є критичними з точки зору працездатності жорсткого диску?
3. Які основні можливості програми HDDExpert?
4. Які основні можливості програми HD Tune?
5. Які види тестів можна виконати за допомогою програми HD Tune?

## Лабораторна робота 3

### ТЕСТУВАННЯ ПК ПРОГРАМОЮ «AIDA64»

**Мета роботи:** Вивчити і протестувати устаткування комп'ютера, записати його характеристики за допомогою програми AIDA64.

#### Хід роботи

#### 1. Основні теоретичні відомості. Призначення програмного комплексу AIDA64

За допомогою програми AIDA64 можна отримати найдетальнішу інформацію про комп'ютер. Ця програма докладно повідомляє все про процесор, материнську плату, відеокарту, мережевий адаптер, накопичувачі, пристрої вводу та ін. (рис. 3.1).

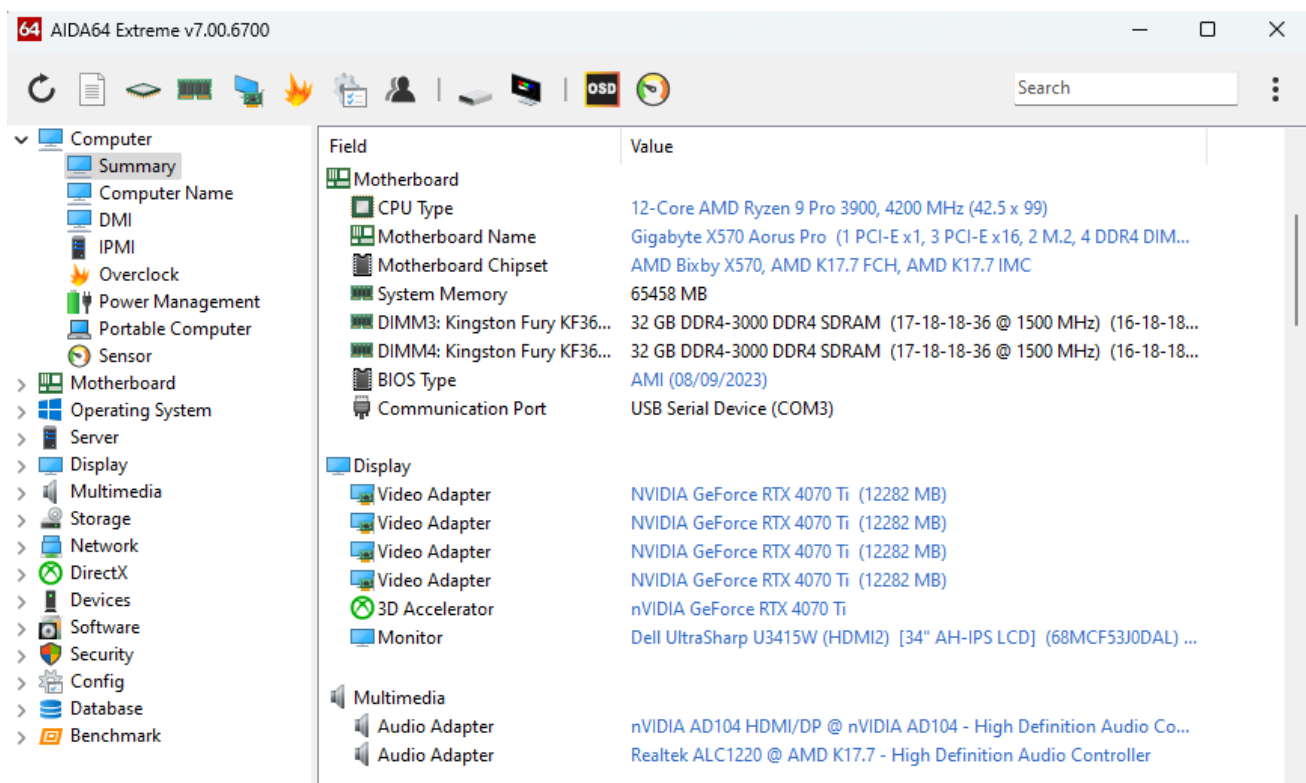


Рисунок 3.1 – Робоче вікно AIDA64

З розгорнутого списку функціональних можливостей AIDA64 Extreme Edition треба виділити такі: виявлення обладнання, інструментальний моніторинг,

екранний моніторинг і сповіщення, тестування продуктивності системи і стрес-тести обладнання, інформація про програмне забезпечення.

Програма здатна визначати тип флеш-пам'яті, модель контролера і швидкість передачі даних. Обсяг наданих AIDA64 даних значний – доступ до них забезпечується з меню розділів, які об'єднують основні модулі програми.

Почерпнути інформацію про мережеві адаптери можна в розділі «Мережа», а про шини, порти, клавіатури, миші та ін. – у розділі «Пристрої». Крім того, з меню «Сервіс» відкривається панель AIDA64 CPUID (рис. 3.2), на якій у компактному вигляді відображаються дані про процесор, материнську плату, пам'ять і чіпсет.



Рисунок 3.2 – Панель AIDA64 CPUID

AIDA64 також містить інструмент «Monitor Diagnostics», що дозволяє відкалібрувати LCD або CRT дисплеї за допомогою різних тестів.

В розділі тестування можна отримати базовий тест стабільності системи. Під час цієї процедури (в розділі меню «Сервіс» обрати «Тест стабільності системи») у реальному часі відслідковуються показники температури і напруги, швидкості обертання кулера та ін. Також можна тестувати обладнання окремо: спочатку процесор з математичним співпроцесором («Stress CPU» + «Stress FPU»), потім пам'ять і кеш («Stress system memory» + «Stress cache») та ін.

## **2. Індивідуальні завдання**

1. Зліва у вікні меню програми AIDA64, у списку «Комп'ютер» вибрати пункт «Сумарна інформація». Скласти звіт по сумарній інформації.

2. Отримати інформацію про центральний процесор ПК. Для цього зліва у вікні меню в списку «Системна плата» вибрати пункт «ЦП». Скласти звіт за основними властивостями ЦП.

3. Дослідження материнської (системної) плати ПК. У вікні меню в списку «Системна плата» вибрати пункт «Системна плата». Скласти звіт за основними властивостями системної плати.

4. Ознайомитися з чіпсетом материнської плати. У вікні меню в списку «Системна плата» вибрати пункт «чіпсет».

5. Ознайомитися з системою зберігання даних ПК ПЗП. У вікні меню в списку «Зберігання даних» вибрати пункт «Зберігання даних Windows», після чого в правому верхньому вікні з'явиться список всіх можливих ПЗП досліджуваного комп'ютера.

6. Ознайомитися з пристроями відображення інформації. Для цього в розділі «Відображення» вибрати послідовно пункти «Відео Windows», «Графічний процесор» і «Монітор».

7. Ознайомитися з наявними на платі портами вводу-виводу.

8. Провести тестування швидкодії ОЗП. Для цього перейти в розділ «Тест» і вибрати відповідні пункти.

### **3. Зміст звіту**

1. Тема та мета роботи.
2. Результати виконання завдань.
3. Висновки.

#### **Контрольні запитання**

1. Як отримати інформацію про датчики, розташовані в ПК?
2. Як отримати сумарну інформацію про ПК?
3. Як отримати інформацію про ім'я комп'ютера?
4. Як отримати інформацію про DMI та IPMI?
5. Як провести стресове тестування ПК?

## Лабораторна робота 4

### ТЕСТУВАННЯ ПК ПРОГРАМОЮ «SISOFTWARE SANDRA»

**Мета роботи:** Вивчити і протестувати устаткування комп'ютера, записати його характеристики за допомогою програми SiSoftware Sandra.

#### Хід роботи

### 1. Основні теоретичні відомості

#### 1.1 Призначення програми SiSoftware Sandra

SiSoftware Sandra (скорочення від System Analyzer, Diagnostic and Reporting Assistant) – це інформаційна і діагностична програма, яка надає практично всю необхідну користувачеві інформацію про апаратне і програмне забезпечення комп'ютера. Розробник програми – компанія SiSoftware. Програма детально аналізує конфігурацію комп'ютера і периферію, виконує тести потужності системи і окремих компонентів, а після завершення тестів рекомендує конкретні дії для підвищення загальної продуктивності. Як і більшість програм, які працюють у середовищі Windows, робоче вікно SiSoftware Sandra (рис. 4.1) містить рядок меню, панель інструментів і рядок стану. Інший простір робочого вікна займають значки Майстрів (Wizard Modules) і Інформаційних Модулів (Information Modules).

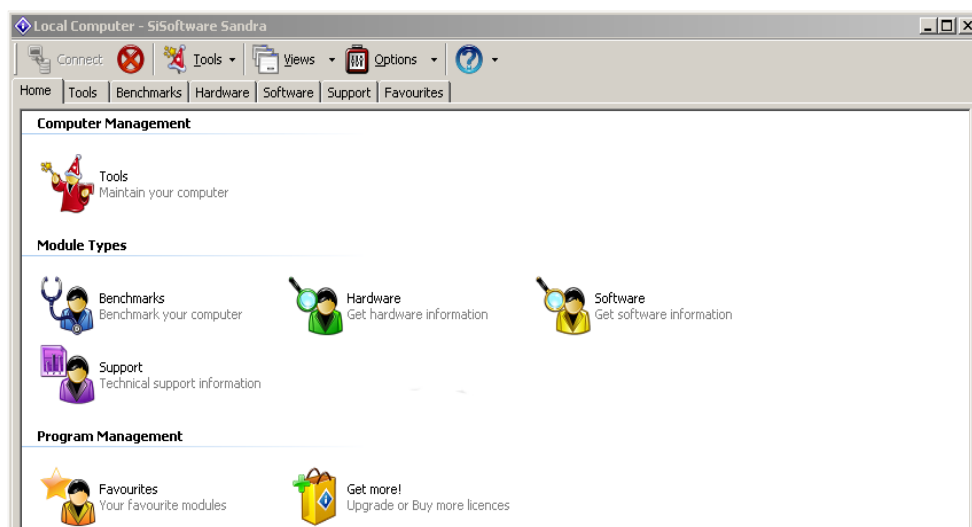


Рисунок 4.1 – Робоче вікно SiSoftware Sandra

Кожен Майстер (Wizard) дозволяє виконати низку певних задач. Наприклад, Майстер узагальненого індексу продуктивності (Combined Performance Index Wizard) допомагає швидко порівняти всі індекси продуктивності, тобто результати тестів системи, з еталонними тестами. Майстер збільшення продуктивності (Performance Tune-Up Wizard) виконує всі тести і поєднує в один перелік поради, помилки та попередження.

Модулі в SiSoftware Sandra поділені на чотири основні групи.

1. Інформаційні модулі (Information Modules) – відображають детальні відомості про конкретний пристрій або компонент комп'ютера.

2. Бенчмаркінгові модулі (Benchmarking Modules) – виконують і відображають тести продуктивності різноманітних компонентів системи. Наприклад, Арифметичний тест процесора (CPU Arithmetic Benchmark) визначає продуктивність процесора при виконанні простих арифметичних операцій, модуль Тест кеш і пам'яті (Cache & Memory Benchmark) порівнює продуктивність кеша процесора і підсистеми пам'яті з еталонними зразками.

3. Тестові модулі (Testing Modules) – відображають різноманітні апаратні функції, їх налаштування і переліки обладнання, що використовують ці функції. Наприклад, модуль Встановлення IRQ (Hardware IRQ Settings) відображає перелік запитів апаратних переривань (IRQ), їх налаштування і перелік обладнання, що використовує кожне переривання.

4. Оглядові модулі (Listing Modules) – демонструють зміст системних файлів ініціалізації.

На панелі інструментів є можливість вибору способу відображення інформації в робочому вікні, а також фільтрації відображення модулів залежно від їх функціональності. Найпростіший спосіб визначити склад обладнання комп'ютера і ознайомитись з загальною інформацією про нього – запустити модуль Зведена Інформація (System Summary). Для цього необхідно двічі натиснути клавішею миші на значку Зведена Інформація (System Summary) у групі Інформаційні модулі (Information Modules). Програма починає виконувати аналіз системи, після чого на екрані відобразиться зведена інформація.

## 1.2 Арифметичний тест продуктивності процесора

Для оцінки продуктивності процесора призначені два модуля – Арифметичний тест процесора (CPU Arithmetic Benchmark) і Мультимедійний тест процесора (CPU Multi-Media Benchmark) з групи Benchmarking Modules. Більшість тестів продуктивності SiSoftware Sandra характеризують швидкість виконання тих або інших задач. Зокрема, арифметичний тест процесора визначає швидкість виконання арифметичних обчислень з цілими числами і числами з плаваючою точкою. Але оцінити продуктивність на основі отриманих результатів можна тільки в порівнянні з такими ж характеристиками інших комп'ютерів. У діалозі відображаються кольорові діаграми з результатами тестів для чотирьох еталонних процесорів. Перед виконанням арифметичного тесту процесору бажано вибрати еталонний процесор, з яким будуть порівнюватись результати тестування. Після завершення тесту його результати відображуються як показано на рис. 4.2.

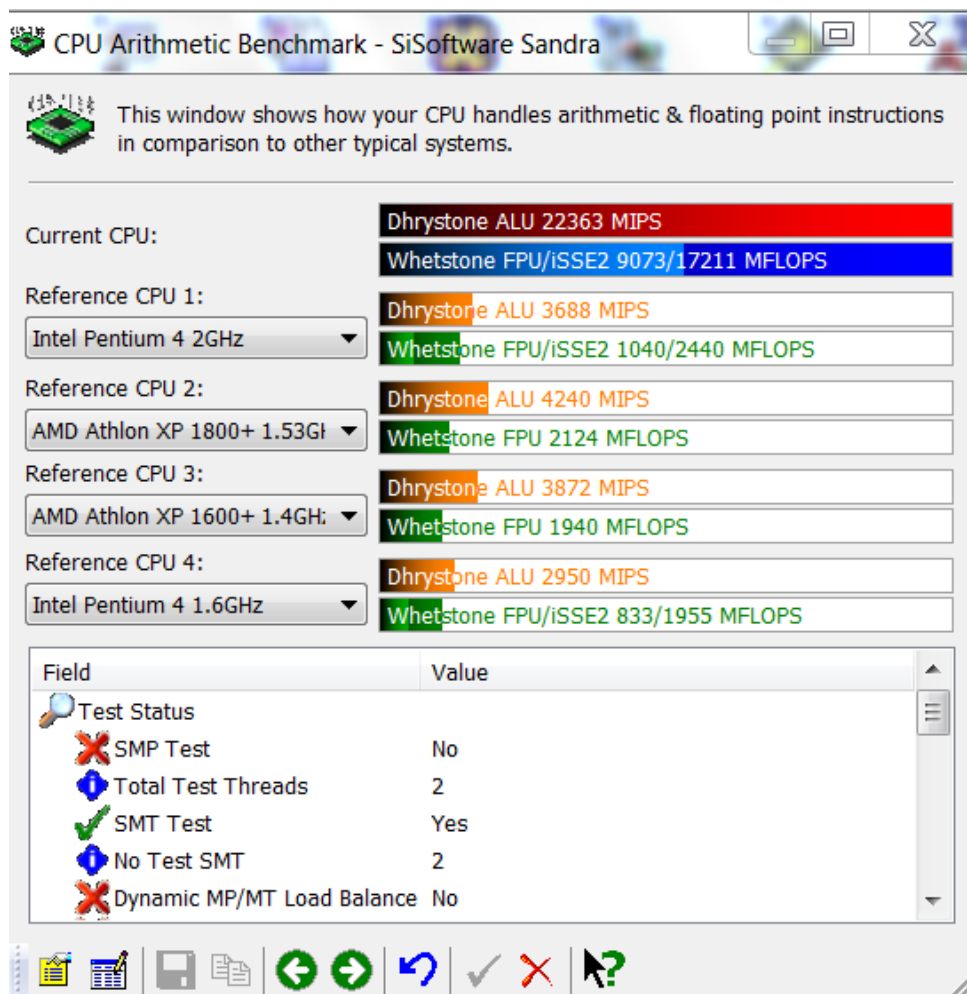


Рисунок 4.2 – Результати арифметичного тесту процесора

SiSoftware Sandra оцінює продуктивність процесора згідно трьох тестів: Dhrystone ALU (ALU – Arithmetic and Logic Unit – Арифметико-логічний пристрій), Whetstone FPU (FPU – Floating Point Unit – Пристрій обробки даних з плаваючою точкою) і Whetstone JSSE2.

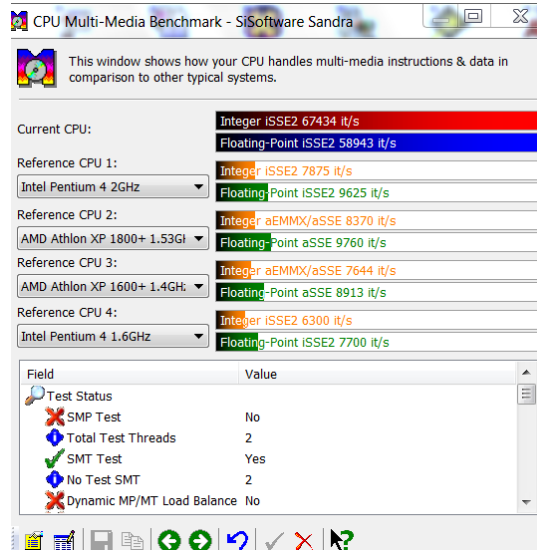


Рисунок 4.3 – Мультимедійний тест процесора

### 1.3 Тест пропускної здатності пам'яті

Для запуску цього модуля треба двічі клацнути мишею на значку модуля Тест пропускної здатності пам'яті (Memory Bandwidth Benchmark). На екрані з'явиться діалог Тест пропускної здатності пам'яті – SiSoftware Sandra (Memory Bandwidth Benchmark – SiSoftware Sandra) (рис. 4.4).

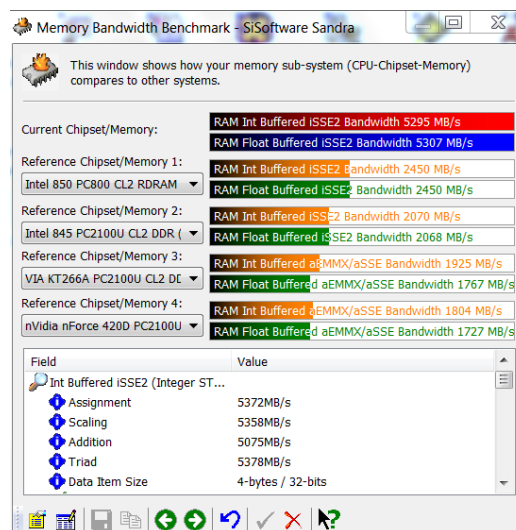


Рисунок 4.4 – Діалог Тест пропускної здатності пам'яті

Тест пропускної здатності пам'яті заснований на відомому тесті STREAM. Результат цього тесту відображається у вигляді двохлінійних діаграм: для цілих чисел (Int) і для даних з плаваючою точкою (Float). На кожній діаграмі вказується використаний метод тестування, наприклад Buff'd (буферизація), набір інструкцій – MMX, EMMX, SSE, SSE2 і пропускна здатність

## **2. Індивідуальні завдання**

1. Запустити програму SiSoftware Sandra.
2. Відкрити вікна «Зведена інформація», «Інформація про материнську плату».
3. Занести в звіт дані про систему, процесор, материнську плату і чіпсет ПК.
4. Занести в звіт результати арифметичного та мультимедійного тестів процесора.
5. Занести в звіт результати тесту пропускної здатності пам'яті.

## **3. Зміст звіту**

1. Тема та мета роботи.
2. Результати виконання завдань.
3. Висновки.

## **Контрольні запитання**

1. Які задачі вирішує програма SiSoftware Sandra?
2. З яких модулів складається робоче вікно SiSoftware Sandra?
4. Які компоненти знаходяться у вікні Інформація про материнську плату?
5. Які результати показує Арифметичний тест процесора?
6. Які результати показує Тест пропускної здатності пам'яті?

## Лабораторна робота 5

### ВИЗНАЧЕННЯ КІЛЬКІСНИХ ХАРАКТЕРИСТИК НАДІЙНОСТІ ЗА СТАТИСТИЧНИМИ ДАНИМИ ПРО ВІДМОВИ ЕЛЕМЕНТІВ

**Мета роботи:** Отримати практичні навички реалізації в середовищі програмування прикладних задач теорії надійності.

#### Хід роботи

##### 1. Основні теоретичні відомості

Надійність – властивість об’єкта зберігати в часі у встановлених межах значення всіх параметрів, що характеризують здатність виконувати необхідні функції в заданих режимах і умовах застосування, технічного обслуговування, ремонту, зберігання і транспортування.

Безвідмовність – властивість об’єкта безупинно зберігати працездатність протягом деякого часу або деякого напрацювання.

Ремонтопридатність – властивість об’єкта, що полягає в пристосованості його до попередження і виявлення відмов та відновлення працездатності об’єкта або шляхом проведення ремонту, або шляхом заміни комплектуючих елементів, які відмовили. Розрізняють дві самостійні характеристики ремонтпридатності: пристосованість до проведення ремонту і пристосованість до заміни в процесі експлуатації.

Довговічність – властивість об’єкта зберігати працездатність до настання граничного стану, тобто такого стану, коли об’єкт має бути спрямований або в ремонт (середній або капітальний), або вилучений з експлуатації.

Працездатність – такий стан об’єкта, при якому він здатний виконувати задані функції, задовольняючи вимогам нормативно-технічної документації. Працездатність – характеристика стану об’єкта в певний момент часу.

Надійність ділиться на апаратну, програмну, експлуатаційну і функціональну.

Відмова об’єкта – подія, що полягає в тому, що об’єкт або повністю, або частково втрачає властивість працездатності. При повній втраті працездатності

виникає повна відмова, при частковій – часткова. Раптовій відмові може передувати поступове накопичення пошкоджень. Поступова відмова виникає головним чином внаслідок зносу і старіння матеріалів.

Відмова може бути короткочасною або самоусуваною. У цьому випадку вона називається збоєм. Відмови доцільно поділяти на апаратні та програмні.

Апаратною відмовою прийнято вважати подію, при якій виріб втрачає працездатність і для його відновлення потрібне проведення ремонту апаратури або заміна відмовленого виробу на справний.

Програмною відмовою вважається подія, при якій об'єкт втрачає працездатність через недосконалість програми. Характерною ознакою програмної відмови є те, що вона усувається шляхом виправлення програми.

У безпосередньому зв'язку з поняттям «надійність» знаходиться поняття «ефективність». Ефективністю об'єкта називається властивість об'єкта видавати деякий корисний результат (ефект) при використанні його за призначенням.

Надійність – це величина випадкова, тому для вивчення надійності використовується теорія ймовірності. Надійність завжди співвідноситься зі статистичними даними про роботу об'єкта. Тому, для обчислення сторін надійності потрібна вибірка, що містить відомості про роботу системи на певному інтервалі часу.

Ймовірність безвідмовної роботи за статистичними даними про відмови оцінюється виразом:

$$P^*(t) = \frac{n(t)}{N}, \quad (5.1)$$

де  $n(t)$  – число виробів, які не відмовили до моменту часу  $t$ ;  $N$  – число виробів, поставлених на випробування;  $P^*(t)$  – статистична оцінка ймовірності безвідмовної роботи виробу.

Для ймовірності відмови за статистичними даними:

$$q^*(t) = \frac{N - n(t)}{N}, \quad (5.2)$$

де  $N - n(t)$  – число виробів, які відмовили до моменту часу  $t$ ;  $q^*(t)$  – статистична оцінка ймовірності відмови виробу.

Ймовірність відмови – це величина, зворотна величині безвідмовної роботи. Частота відмов за статистичними даними про відмови визначається виразом:

$$f^*(t) = \frac{\Delta n(t)}{N \cdot \Delta t}, \quad (5.3)$$

де  $\Delta n(t)$  – число виробів, які відмовили, на інтервалі  $[t, t + \Delta t]$ ;  $f^*(t)$  – статистична оцінка частоти відмов виробу;  $\Delta t$  – інтервал часу.

Частота відмов відображає середнє число виробів, які відмовили в одиницю часу на інтервалі  $[t, t + \Delta t]$ . Інтенсивність відмов за статистичними даними про відмови визначається формулою:

$$\lambda^*(t) = \frac{\Delta n(t)}{\Delta t \cdot n(t)}, \quad (5.4)$$

де  $n(t)$  – число виробів, які не відмовили до моменту часу  $t$ ;  $\Delta n(t)$  – число виробів, які відмовили, на ділянці часу  $[t, t + \Delta t]$ ;  $\lambda^*(t)$  – статистична оцінка інтенсивності відмов виробу.

Середній час безвідмовної роботи виробу за статистичними даними оцінюється виразом:

$$m_t^* = \frac{1}{N} \sum_{i=1}^N t_i, \quad (5.5)$$

де  $t_i$  – час безвідмовної роботи  $i$ -го виробу;  $N$  – загальне число виробів, поставлених на випробування;  $m_t^*$  – статистична оцінка середнього часу безвідмовної роботи виробу.

Для визначення  $m_t^*$  необхідно знати моменти виходу з ладу всіх  $N$  виробів. Можна визначати  $m_t^*$  за рівнянням:

$$m_t^* \approx \sum_{i=1}^m n_i t_{\text{сер}i}, \quad (5.6)$$

де  $n_i$  – кількість виробів, які вийшли з ладу, в  $i$ -му інтервалі часу;  $t_{\text{сеп}_i} = (t_{i-1} + t_i)/2$ ;  
 $m = t_k / \Delta t$ ;  $\Delta t = t_{i+1} - t_i$ ;  $t_{i-1}$  – час початку  $i$ -го інтервалу;  $t_i$  – час кінця  $i$ -го інтервалу;  $t_k$   
– час, протягом якого вийшли з ладу всі вироби;  $\Delta t$  – інтервал часу.

## 2. Індивідуальні завдання

Розробити програму, яка виконує розрахунки згідно індивідуального завдання. Передбачити можливість дослідження зміни результатів залежно від вихідних даних.

1. На випробування поставлено 100 однотипних виробів. За 4000 год. відмовило 50 виробів. За інтервал часу [4000...4100 год.] відмовило ще 20 виробів. Потрібно визначити  $f^*(t)$ ,  $\lambda^*(t)$  при  $t = 4000$  год.

2. На випробування поставлено 100 однотипних виробів. За 4000 год. відмовило 50 виробів. Потрібно визначити  $p^*(t)$  і  $q^*(t)$  при  $t = 4000$  год.

3. Протягом 1000 год. з 10 гіроскопів відмовило два гіроскопи. За інтервал часу [1000...1100 год.] відмовив ще один гіроскоп. Потрібно визначити  $f^*(t)$ ,  $\lambda^*(t)$  при  $t = 1000$  год.

4. На випробування поставлено 1000 однотипних електронних ламп. За перші 3000 час. відмовило 80 ламп. За інтервал часу [3000...4000 год.] відмовило ще 50 ламп. Потрібно визначити  $p^*(t)$  і  $q^*(t)$  при  $t = 4000$  год.

5. На випробування поставлено 45 шт. виробів. За час  $t = 60$  год. вийшло з ладу 35 виробів. За наступний інтервал часу [60...65 год.] вийшло з ладу ще 3 вироби. Необхідно обчислити  $p^*(t)$  при  $t = 60$  год. та  $t = 65$  год.;  $f^*(t)$ ,  $\lambda^*(t)$  при  $t = 60$  год.

6. На випробування поставлено 8 однотипних виробів. Отримані наступні значення  $t_i$  ( $t_i$  – час безвідмовної роботи  $i$ -го виробу):  $t_1 = 560$  год.;  $t_2 = 700$  год.;  $t_3 = 800$  год.;  $t_4 = 650$  год.;  $t_5 = 580$  год.;  $t_6 = 760$  год.;  $t_7 = 920$  год.;  $t_8 = 850$  год. Визначити статистичну оцінку середнього часу безвідмовної роботи виробу.

7. На випробування поставлено 1000 виробів. За час  $t = 11000$  год. вийшло з ладу 410 виробів. За наступний інтервал часу [11000...12000 год.] вийшло з ладу

ще 40 виробів. Необхідно обчислити  $p^*(t)$  при  $t = 11000$  год. і  $t = 12000$  год., а також  $f^*(t)$ ,  $\lambda^*(t)$  при  $t = 11000$  год.

### **3. Зміст звіту**

1. Тема та мета роботи.
2. Результати виконання завдань.
3. Висновки.

### **Контрольні запитання**

1. Що таке безвідмовність?
2. У чому полягає відмова об'єкту?
3. Як визначається ймовірність відмови?
4. Як визначається частота відмов?
5. Як визначається інтенсивність відмов?

## Лабораторна робота 6

### ЗАВАДОСТІЙКЕ КОДУВАННЯ. КОДИ ХЕМІНГА

**Мета роботи:** Вивчити способи кодування інформації, що дозволяють виявляти і виправляти помилки.

#### Хід роботи

##### 1. Основні теоретичні відомості

В реальних умовах прийом двійкових символів часто відбувається з помилками, коли замість символу "1" приймається символ "0" і навпаки. Помилки можуть виникати через перешкоди, що діють у каналі зв'язку, зниження рівня передачі та ін. Загальноприйнятим критерієм оцінки якості передачі в дискретних каналах є допустима ймовірність помилки для даного виду повідомлень. Так, допустима ймовірність помилки при телеграфного зв'язку може становити  $10^{-3}$  (на знак), а при передачі даних – не більше  $10^{-6}$  (на символ). Для забезпечення таких значень ймовірностей одного поліпшення тільки якісних показників каналу зв'язку може виявитися недостатнім. Тому основним заходом є застосування спеціальних методів підвищення якості прийому інформації. Ці методи можна розбити на дві групи:

– методи збільшення завадостійкості прийому одиничних елементів (символів) дискретної інформації, пов'язані з вибором рівня сигналу, відношення сигнал-перешкода (енергетичні характеристики), ширини смуги каналу, методів прийому та ін.,

– методи виявлення і виправлення помилок, засновані на штучному введенні надмірності в передане повідомлення.

Практичні можливості збільшення надмірності за рахунок потужності і ширини спектра сигналу в системах передачі дискретної інформації по стандартних каналах різко обмежені. Тому для підвищення якості прийому йдуть по шляху збільшення часу передачі і використовують такі основні способи:

- 1) багаторазова передача кодових комбінацій (метод повторення);
- 2) одночасна передача кодової комбінації з кількох паралельно працюючих каналів;

3) перешкодостійке (коригуюче) кодування.

Найбільш доцільно надмірність використовується при застосуванні завадостійких (коригувальних) кодів.

Коди Хемінга – це одні з найбільш поширених систематичних кодів, які виправляють помилки. До кодів Хемінга належать коди з мінімальною кодовою відстанню  $d_{min}=3$ , що виправляють всі поодинокі помилки.

Формування  $r$  перевірочних елементів у комбінаціях цих кодів виконують по  $k$  інформаційних елементах. Таким чином, довжина кодової комбінації  $n = k + r$ . Перевірочними елементами є лінійні комбінації інформаційних елементів, тобто зважені суми інформаційних елементів з ваговими коефіцієнтами 1 і 0. Послідовність одиниць і нулів у кодовій комбінації називається кодовим вектором. Кодам Хемінга притаманні властивості лінійних кодів: сума (різниця) векторів лінійного коду дає вектор, який належить цим кодам; лінійні коди утворюють алгебраїчну групу щодо операції додавання за модулем «2»; мінімальна кодова відстань між векторами групового коду дорівнює мінімальній вазі ненульових кодових векторів. При передачі кодового вектора може бути спотворений будь-який елемент, кількість таких ситуацій дорівнює  $n$ . До цього слід додати ще одну ситуацію, коли помилка не виникає. Таким чином, загальна кількість  $2^r$  комбінацій перевірочних елементів може перевищувати кількість можливих помилкових ситуацій у коді з урахуванням відсутності помилок і визначення місця помилки:

$$2^r \geq n + 1. \quad (6.1)$$

Оскільки  $2^n = 2^{k+r} = 2^k \cdot 2^r$ , можна записати

$$2^n \geq (n + 1) \cdot 2^k, \quad (6.2)$$

де  $2^n$  – повна кількість комбінацій коду.

Мінімальне співвідношення коригуючих та інформаційних розрядів, менше якого код не може зберігати задані коригуючі властивості, визначається виразом:

$$2^r - 1 = n. \quad (6.3)$$

Для розрахунку основних параметрів кодів Хемінга можна задати кількість перевірочних елементів  $r$ , тоді з останнього виразу визначається  $n$ , а кількість інформаційних елементів  $k = n - r$ . Співвідношення між  $r$ ,  $n$  та  $k$  для кодів Хемінга наведено в табл. 6.1.

Таблиця 6.1

$k$	1	1	2	3	4	4	5	6	7	8	9	10	11	11
$r$	2	3	3	3	3	4	4	4	4	4	4	4	4	5
$n$	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Характерна особливість перевірочної матриці коду з  $d_{min} = 3$  складається в тому, що її стовпці є різними ненульовими комбінаціями довжиною  $r$ .

Наприклад, при  $r = 4$ ,  $n = 15$  перевірочна матриця може мати такий вигляд:

$$H_{(15,4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \end{bmatrix} \quad (6.4)$$

Таким чином, якщо взяти комбінації чотирьохелементного двійкового простого коду і відкинути нульову комбінацію, можна досить легко отримати перевірочну матрицю, записавши всі кодові комбінації послідовно в стовпці матриці  $H$ .

Після перестановки стовпців, які мають одну одиницю, матриця (6.4) набуває вигляду:

$$H_{(15,4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & b_1 & b_2 & b_3 & b_4 \end{bmatrix} \quad (6.5)$$

Відповідно до матриці (6.5) отримуємо систему перевірочних рівнянь, за допомогою яких знаходимо перевірочні розряди:

$$\left. \begin{aligned} b_1 &= a_5 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{11}; \\ b_2 &= a_2 \oplus a_3 \oplus a_4 \oplus a_8 \oplus a_9 \oplus a_{10} \oplus a_{11}; \\ b_3 &= a_1 \oplus a_3 \oplus a_4 \oplus a_6 \oplus a_7 \oplus a_{10} \oplus a_{11}; \\ b_4 &= a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7 \oplus a_9 \oplus a_{11}. \end{aligned} \right\} \quad (6.6)$$

Поява помилки в кодовій комбінації веде до невиконання тих перевірочних співвідношень (6.6), в які входить значення помилкового розряду.

Так, якщо помилка виникла в сьомому інформаційному розряді ( $a_7$ ), то не виконуються перше, третє і четверте співвідношення (6.6), тобто синдром дорівнює 1011 і збігається з сьомим стовпчиком матриці  $H$  (6.5). Таким чином, місцезнаходження стовпця матриці  $H$ , що збігається зі знайденим синдромом, визначає місце помилки.

Обчислення значення синдрому обов'язково збігається з одним з стовпців матриці  $H$  (стовпці вибирають всі можливі  $r$ -розрядні двійкові комбінації). Хемінг запропонував розташувати стовпці перевірочної матриці так, щоб номер  $i$ -го стовпця матриці  $H$  і номер розряду кодової комбінації відповідали бінарному подання числа  $i$ . Тоді синдром, який знайдено з перевірочних рівнянь, буде двійковим поданням номера розряду кодової комбінації, в якому виникла помилка. Для цього перевірочні розряди повинні знаходитися не в кінці кодової комбінації, а на номерах позицій, які визначаються ступенем двійки ( $2^0, 2^1, 2^2, \dots, 2^{2r-1}$ ) як у матриці (6.4), тому що кожен з них входить тільки в одне з перевірочних рівнянь. В останньому випадку перевірочні розряди розміщуються між інформаційними.

Відповідно, синдром перевірочної матриці (6.4) визначається з системи рівнянь:

$$\left. \begin{aligned} S_1 &= u_1 \oplus u_3 \oplus u_5 \oplus u_7 \oplus u_9 \oplus u_{11} \oplus u_{13} \oplus u_{15}; \\ S_2 &= u_2 \oplus u_3 \oplus u_6 \oplus u_7 \oplus u_{10} \oplus u_{11} \oplus u_{14} \oplus u_{15}; \\ S_3 &= u_4 \oplus u_5 \oplus u_6 \oplus u_7 \oplus u_{12} \oplus u_{13} \oplus u_{14} \oplus u_{15}; \\ S_4 &= u_8 \oplus u_9 \oplus u_{10} \oplus u_{11} \oplus u_{12} \oplus u_{13} \oplus u_{14} \oplus u_{15}. \end{aligned} \right\} \quad (6.7)$$

Як перевірочні вибираються розряди  $u_1, u_2, u_4, u_8$ , які зустрічаються в системі рівнянь (6.7) по одному разу.

Наприклад, якщо необхідно закодувати повідомлення 11001010110 двійкового простого коду ( $k = 11$ ) в кодї Хемінга, то потрібно визначити перевірочні розряди в комбінації:

$$u_1 u_2 1 u_4 1 0 0 u_8 1 0 1 0 1 1 0.$$

З перевірконої матриці (6.4) маємо:

$$\left. \begin{aligned} u_1 &= u_3 \oplus u_5 \oplus u_7 \oplus u_9 \oplus u_{11} \oplus u_{13} \oplus u_{15}; \\ u_2 &= u_3 \oplus u_6 \oplus u_7 \oplus u_{10} \oplus u_{11} \oplus u_{14} \oplus u_{15}; \\ u_4 &= u_5 \oplus u_6 \oplus u_7 \oplus u_{12} \oplus u_{13} \oplus u_{14} \oplus u_{15}; \\ u_8 &= u_9 \oplus u_{10} \oplus u_{11} \oplus u_{12} \oplus u_{13} \oplus u_{14} \oplus u_{15}. \end{aligned} \right\} \quad (6.8)$$

Обчислюючи  $u_1, u_2, u_4$  та  $u_8$  у відповідності з (6.8), отримуємо:

$$\left. \begin{aligned} u_1 &= 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1; \\ u_2 &= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1; \\ u_4 &= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1; \\ u_8 &= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 0. \end{aligned} \right\} \quad (6.9)$$

Отже, комбінація коду Хемінга для розглянутого повідомлення матиме вигляд 111110001010110.

Тепер припустимо, що шостий елемент цієї комбінації приймається помилково, тобто отримане повідомлення 111111001010110. Обчислюючи синдром за допомогою системи рівнянь знаходимо  $S_1=0; S_2=1; S_3=1; S_4=0$ , тобто синдром має вигляд 0110. Якщо це двійкове число перевести в десяткове, то отримаємо 6. Таким чином, необхідно виправити шостий розряд повідомлення, яке надійшло.

## 2. Індивідуальні завдання

1. Побудувати перевірочну матрицю двійкового коду Хемінга для передачі заданої інформації ( $N_{10}$  – згідно варіанту по табл. 6.2).

2. За допомогою побудованої перевірконої матриці закодувати число, представлене в десятковій формі.

3. Показати виправлення одноразової помилки в заданому розряді отриманої комбінації коду.

Таблиця 6.2

№ вар.	$N_{10}$	№ розряду, який передається з помилкою	№ вар.	$N_{10}$	№ розряду, який передається з помилкою
1	32111	1	11	8411	11
2	21793	2	12	16513	12
3	9695	3	13	32815	13
4	797	4	14	32112	1
5	1299	5	15	21794	2
6	1671	6	16	9696	3
7	2293	7	17	798	4
8	2675	8	18	1290	5
9	4667	9	19	1672	6
10	5519	10	20	2294	7

### 3. Зміст звіту

1. Тема та мета роботи.
2. Навести розрахунки згідно з індивідуальним завданням.
3. Висновки.

### Контрольні запитання

1. Поясніть методи підвищення якості прийому інформації, яка передається.
2. Назвіть переваги і недоліки методу багаторазового повторення.
3. У чому полягає сутність завадостійкого кодування?
4. Які завдання вирішують перешкодостійкі коди?
5. У чому полягає складність вибору коригуючого коду для реальних каналів зв'язку?
6. Наведіть приклади рівномірних і нерівномірних кодів.
7. Які коди називаються систематичними? Які їх основні властивості?
8. Перерахуйте основні характеристики коригувальних кодів.
9. Що таке мінімальна кодова відстань?
10. Дайте визначення синдрому помилок.

## Лабораторна робота 7

### ДІАГНОСТИКА МЕРЕЖ WINDOWS ВБУДОВАНИМИ ПРОГРАМАМИ

**Мета роботи:** ознайомитись з вбудованими до Windows діагностичними програмами та навчитись їх використовувати.

#### Хід роботи

##### 1. Основні теоретичні відомості

В комп'ютерній мережі передача інформації від одного вузла (комп'ютера) до іншого здійснюється за допомогою пакетів, які можна розглядати як набір бітів службової інформації (адреса вузла джерела пакета, адреса вузла призначення, довжина пакета та ін.) та інформації для передачі. Для передачі пакетів використовується адресація TCP/IP (Transmission Control Protocol / Internet Protocol – протокол управління передачею / міжмережевий протокол). TCP/IP є набором протоколів, які забезпечують зв'язок комп'ютерів у мережі Internet. Сімейство TCP/IP включає в себе протокол контролю транспортування (TCP), адресний протокол Internet (IP) і безліч інших протоколів. IP-адреса – це 32-розрядне значення, яке використовується для правильної ідентифікації джерела і адреси пункту призначення. Адреса IP зазвичай представляється в наступному вигляді: 204.107.2.100. Адресу можна розбити на чотири позиції по вісім бітів кожна. Ці позиції називають октетами. У наведеному прикладі IP-адреси число 204 – це значення першого октету, а 100 – четвертого. Наведена в прикладі IP-адресу можна легко записати в двійковому вигляді: 11001100.01101011.00000010.01100100.

IP-адреса складається з двох частин: номера мережі і номера вузла. Номер мережі може бути обраний адміністратором мережі довільно, або призначений за рекомендацією спеціального підрозділу Internet (Network Information Center, NIC), якщо мережа повинна працювати як складова частина Internet. Зазвичай провайдери послуг Internet отримують діапазони адрес у підрозділах NIC, а потім розподіляють їх між своїми абонентами. Номер вузла в протоколі IP призначається незалежно від локальної адреси вузла. Розподіл IP-адреси на поле номера мережі і номера вузла – гнучке, і межа між цими полями може встановлюватися досить

довільно. Вузол може входити в кілька IP-мереж, у цьому випадку вузол повинен мати кілька IP-адрес, по числу мережевих зв'язків. Таким чином, IP-адреса характеризує не окремий комп'ютер, а одне мережеве з'єднання. Для того, щоб в IP-адресі розрізнити номер мережі і номер вузла використовують маску підмережі. Існують класи А, В, С, D, Е. Класи адрес Internet на прикладі IP-адреси 204.107.2.100 наведені в табл. 7.1. У табл. 7.2 узагальнено відомості по класах і вузлах мережі.

Таблиця 7.1

Клас	Адреса мережі	Адреса вузла	Стандартне значення маски підмережі
А	104	105.2.100	255.0.0.0
В	104.105	2.100	255.255.0.0
С	104.105.2	100	255.255.255.0

Таблиця 7.2

Клас	Кількість можливих мереж	Кількість можливих вузлів у мережі
А	126	16777214
В	16 384	65534
С	2 097 151	254

Крім IP-адреси існує локальна адреса сайту, який визначається технологією, за допомогою якої побудована окрема мережа, в яку входить даний вузол. Для вузлів, що входять у локальні мережі – це MAC-адреса мережного адаптера або порту маршрутизатора (наприклад, 11-A0-17-3E-BF-06). Ці адреси призначаються виробниками устаткування. MAC-адреса має наступний формат: старші 3 байти – ідентифікатор виробника, молодші 3 байти призначаються унікальним чином самим виробником.

Метою усунення несправностей у налаштуванні TCP/IP є відновлення нормальної роботи мережі. Для пошуку несправностей можна використовувати спеціальні діагностичні утиліти. Операційна система Windows має значну кількість вбудованих діагностичних програм для ПК і мереж. Утиліта PING (Packet Internet Groper) є одним з головних засобів, які використовуються для

налагодження мереж. Вона дозволяє перевіряти роботу програм TCP/IP на віддалених ПК, перевіряти адреси пристроїв у локальній мережі та ін. Запити утиліти PING передаються по протоколу ICMP (Internet Control Message Protocol). Одержавши такий запит, програмне забезпечення, яке реалізує протокол IP в адресата, негайно посилає ехо-відповідь. Запити посилають задану кількість разів до тих пір, поки користувач не введе команду переривання, після чого виводяться статистичні дані. Параметри утиліти PING приведені в табл. 7.3.

Таблиця 7.3

Параметр	Призначення
-t	відправлення пакетів на зазначений вузол до команди переривання
-a	визначення IP-адрес по іменах вузлів
-n <i>число</i>	число ехо-запитів
-l <i>розмір</i>	розмір поля даних у відправлених пакетах з ехо-запитом
-f	установка прапорця, який забороняє фрагментацію пакета
-i <i>TTL</i>	завдання часу життя (у секундах) пакета
-v <i>TOS</i>	завдання типу служби в заголовку пакета
-r <i>число</i>	запис маршруту з ехо-запитом для зазначеного числа переходів
-s <i>число</i>	варіант штампа часу в заголовку пакета для запису часу прибуття пакета для кожного переходу
-j <i>списокВузлів</i>	вільний вибір маршруту за списком вузлів
-k <i>списокВузлів</i>	жорсткий вибір маршруту за списком вузлів
-w <i>таймаут</i>	інтервал часу очікування ехо-відповіді

Формат команди:

```
ping [-t] [-a] [-n число] [-l розмір] [-f] [-i TTL] [-v TOS][-r число]
[-s число] [[-j списокВузлів][[-k списокВузлів]][-w таймаут] кінцевеІм'я
```

Утиліта TRACERT призначена для визначення маршруту до точки призначення за допомогою посилки в точку призначення ехо-запитів протоколу Internet Control Message Protocol (ICMP) з різними значеннями строку життя – TTL (Time-To-Live). Таким чином, TRACERT дозволяє виявляти послідовність шлюзів, через які проходить IP-пакет на шляху до пункту свого призначення. Параметри утиліти TRACERT приведені в табл. 7.4. Вихідна інформація являє собою список, який починаючи з першого шлюзу та закінчується пунктом призначення. Крім того, фіксується повний час проходження кожного шлюзу.

Формат команди:

```
tracert [-d] [-h максимальнеЧислоПереходів] [-j списокВузлів]
[-w інтервал] ім'яКінцевогоКомп'ютера
```

Таблиця 7.4

Параметр	Призначення
-d	Забороляє дозвіл IP-адрес проміжних маршрутизаторів в імена
-h <i>максимальнеЧислоПереходів</i>	Задає максимальне число переходів на шляху при пошуку кінцевого вузла
-j <i>списокВузлів</i>	Вказує для повідомлень з ехо-запитом використання параметра вільної маршрутизації в заголовку IP з набором проміжних шлюзів, зазначених у списку вузлів
-w <i>інтервал</i>	Визначає час очікування відповідей протоколу ICMP, які відповідають даному ехо-запиту

## 2. Індивідуальні завдання

За допомогою утиліти PING перевірити стан зв'язку з 7 різними вузлами, вибраними за бажанням. Число запитів, які відправляються, взяти рівним 10. За допомогою утиліти TRACERT визначити маршрути та час проходження пакетів до цих вузлів.

### **3. Зміст звіту**

1. Тема та мета роботи.
2. Результати виконання завдань.
3. Висновки.

#### **Контрольні запитання**

1. Що таке MAC-адреса та маска підмережі?
2. Що таке IP-адреса?
3. На які класи діляться IP-мережі?
4. Практичне застосування утиліти PING.
5. Формат утиліти PING і призначення параметрів.
6. Що таке TTL і для чого воно використовується?
7. Практичне застосування утиліти TRACERT.
8. Формат команди утиліти TRACERT.
9. Особливості спільного використання ключів *-n* та *-t* утиліти PING.

## Лабораторна робота 8

### РОЗРОБКА ПРОГРАМ ДЛЯ ДІАГНОСТИКИ ПАРАМЕТРІВ ПК

**Мета роботи:** ознайомитись з можливістю розробки програм, які дозволяють виконувати аналіз і діагностику ПК.

#### Хід роботи

#### 1. Основні теоретичні відомості

##### 1.1 Робота з системним реєстром ОС Microsoft Windows

Дані про пристрої ПК в ОС Windows зберігаються в системному реєстрі. Наприклад, відомості про BIOS материнської плати зберігаються в ключах реєстру розділу `HKEY_LOCAL_MACHINE \ HARDWARE \ DESCRIPTION \ System`. Для розробки програмного забезпечення, яке виконує аналіз працездатності ПК, можна використовувати будь-яку мову і середовище розробки. Для отримання відомостей про комп'ютерну систему можна використовувати як стандартні функції WinAPI, так і дані, що зберігаються в реєстрі. Розглянемо приклади створення програмного забезпечення, що дозволяє виконувати аналіз і діагностику деяких параметрів ПК. В якості середовища будемо використовувати C++ Builder.

##### 1.2 Визначення параметрів BIOS

Розглянемо приклад програми, що визначає дату і версію BIOS. Продемонструємо роботу з системним реєстром, а саме, виконаємо читання необхідної для цього інформації з відповідних ключів реєстру. Після запуску C++ Builder необхідно розмістити на формі чотири компонента Label, встановивши їх властивості відповідно до табл. 8.1.

Таблиця 8.1

Ім'я компонента	Властивість	Значення
Form1	Caption	Інформація про BIOS
Label1	Caption	Дата:
Label2	Caption	Версія:
Label3	Caption	?
Label4	Caption	?

Вікно форми проєкту прийме вигляд, як показано на рис. 8.1.



Рисунок 8.1 – Вікно форми проєкту визначення параметрів BIOS

Далі необхідно створити обробник події для методу FormCreate, що дозволить виконувати визначення параметрів у процесі запуску даної програми. Приклад вихідного коду має наступний вигляд:

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    TRegistry *reg; // покажчик для роботи з ключами реєстру
    AnsiString S = "Unknown";
    char buf[255];
    int nchar, i;
    reg = new TRegistry;
    // вибираємо кореневий розділ
    reg->RootKey = HKEY_LOCAL_MACHINE;
    // відкриваємо ключ обраного розділу
    if (reg->OpenKey("HARDWARE\\DESCRIPTION\\System",false))
    {
        // виконуємо читання параметрів
        Label3->Caption = reg->ReadString("SystemBiosDate");
        nchar = reg->ReadBinaryData("SystemBiosVersion", buf, 256)-1;
        S = "";
        for (i=0; i<=nchar; i++)
```

```

if (buf[i]==0) S = S + " ";
else S = S + buf[i];
Label4->Caption = S;
// закриваємо відкритий раніше ключ
reg->CloseKey();
}
delete reg;
}

```

В наведеному прикладі для читання даних з системного реєстру використовувався клас TRegistry, для роботи якого необхідно підключити відповідну бібліотеку (#include <registry.hpp>). Вікно робочого проєкту наведено на рис. 8.2.

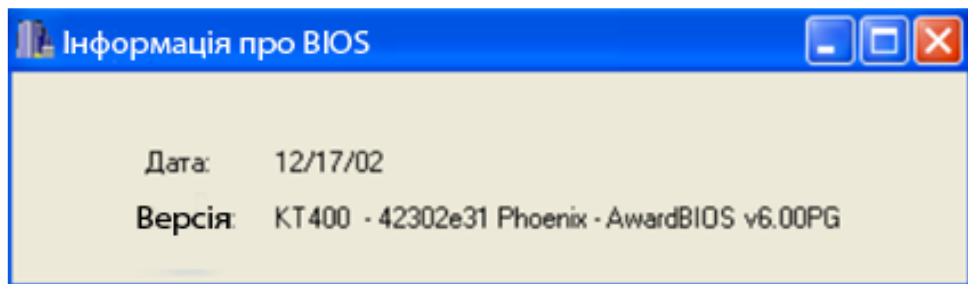


Рисунок 8.2 – Вікно параметрів системи BIOS

### 1.3 Визначення параметрів відеосистеми

Створюємо проєкт і розміщуємо на формі чотири компонента Label, за допомогою яких виводиться вся необхідна інформація. Далі створюється обробник події FormCreate наступного вигляду:

```

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    long int BPP,CP,VR;

```

```

double TC;
AnsiString sBPP;
OSVERSIONINFO OV;
// структура, що зберігає інформацію про версії ОС
// визначення роздільної здатності екрану
Label1->Caption = "Роздільна здатність екрана: "+IntToStr(Screen->Width) +
" x " + IntToStr(Screen->Height) ;
// визначення якості передачі кольору
BPP=GetDeviceCaps(Form1->Canvas->Handle,BITSPIXEL);
CP=GetDeviceCaps(Form1->Canvas->Handle,PLANES);
TC=pow(pow(2,BPP),CP);
switch (BPP)
{
case 4: sBPP = " (16 Color)"; break;
case 8: sBPP = " (256 Color)"; break;
case 16: sBPP = " (High Color)"; break;
case 24, 32: sBPP = " (True Color)"; break;
}
Label2->Caption="Біт на пиксель: "+IntToStr(BPP)+sBPP;
if (TC<3)
{
Label3->Caption="Монохромний дисплей";
}
else
{
Label3->Caption=" Кількість відтінків: "+FloatToStr(TC);
}
// визначення частоти відновлення екрана
OV.dwOSVersionInfoSize=sizeof(OV);
GetVersionEx(&OV);

```

```

if (OV.dwPlatformId==VER_PLATFORM_WIN32_NT)
{
VR=GetDeviceCaps(Form1->Canvas->Handle,VREFRESH);
Label4->Caption=" Частота розгортки: "+IntToStr(VR)+" Гц";
}
else
{
Label4->Caption="Частота розгортки не визначається в даній ОС";
}
}

```

Після запуску вікно програми буде мати вигляд, як показано на рис. 8.3.

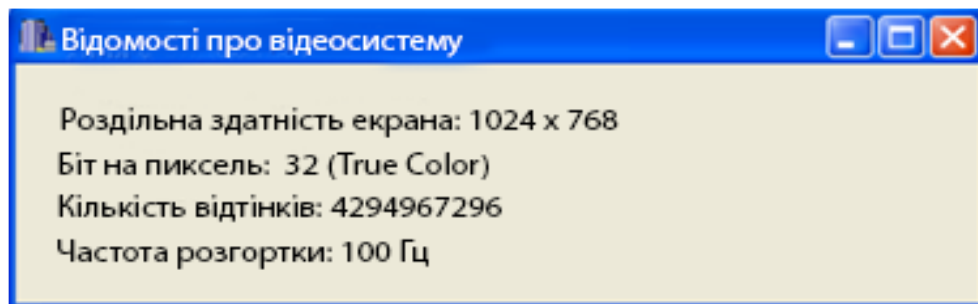


Рисунок 8.3 – Вікно проєкту з інформацією про відеосистему

Для отримання роздільної здатності екрана використовується глобальна змінна Screen. Це покажчик на екземпляр класу TScreen і надає доступ до властивостей Width і Height. Значення властивості Width містить ширину екрана в пікселях, а значення властивості Height – його висоту в пікселях.

```

Label1->Caption = IntToStr(Screen->Width) + " x " +
IntToStr(Screen->Height);

```

Визначення якості передачі кольору і частоти кадрової розгортки

виконується за допомогою Windows API функції GetDeviceCaps, що використовує в якості вхідного параметра контекст графічного пристрою (екран). У даному прикладі використовується звичайна форма VCL для доступу до цього контексту, використовуючи її властивість Canvas.

В якості другого параметру цієї функції використовується іменована ціла константа, залежно від значення якої повертається одно зі значень, пов'язаних з параметрами графічного пристрою. У даному випадку необхідні кольорова роздільна здатність у бітах на піксель і кількість кольорів, що підтримуються системою. Останнє значення залежить від кількості відтінків кольору.

Для обчислення кількості підтримуваних кольорів, необхідно двічі викликати функцію GetDeviceCaps. Один раз для отримання інформації про кольорову роздільну здатність, другий – про кількість кольорових площин.

Функція GetDeviceCaps має наступний вигляд:

```
int GetDeviceCaps
{
HDC hdc, //дескриптор DC
int nIndex //індекс дії
};
```

Для рішення поставленої задачі в якості індексу дії використовуються наступні параметри:

- BITSPixel – кількість суміжних кольорових бітів для кожного пікселя;
- PLANES – кількість кольорових площин;
- VREFRESH – частота кадрової розгортки.

Для ознайомлення з повним переліком параметрів функції можна скористатися довідковою системою.

## **1.4 Інформація про диски**

Інформація про стан диску достатньо актуальна для користувачів ПК.

Нестача дискового простору – одна з найчастіших проблем, з якою приходиться стикатися користувачам.

Під час роботи ОС Windows створює тимчасові файли на диску, а також вивантажує інформацію з оперативної пам'яті на диск у файл підкачування, недостатній обсяг вільного простору може приводити як до сповільнення роботи системи, так і до збоїв системи. Окрім інформації про обсяг, існують й інші параметри, наприклад, серійний номер тома, файлова система та ін.

Для читання інформації про диски можна скористатися наступними вказівками: на форму програми додати компонент DriveComboBox, а також кілька компонентів Label для відображення отриманої інформації про диск. Також необхідно створити подію onChange компонента DriveComboBox1, де виконується виклик функції UpdateDisk:

```
void __fastcall TForm1::DriveComboBox1Change(TObject *Sender)
{
    UpdateDisk();
}
```

Функція UpdateDisk оголошена в розділі public класу TForm1 наступним чином:

```
public: // User declarations
void __fastcall UpdateDisk ();
```

За допомогою цієї функції визначаються наступні параметри диска: ім'я тома, серійний номер. Функція має наступний вигляд:

```
void __fastcall TForm1::UpdateDisk()
{
    DWORD VolSN,MaxCompLen,FSFlags, FC,SPC,BPS,NC;
```

```

char VolName[255],FSName[100];
// формування імені диска
AnsiString S=UpperCase(DriveComboBox1->Drive)+":\\";
Label1->Caption=S;
// виклик функції для читання інформації про диск
if
(GetVolumeInformation(S.c_str(),VolName,255,&VolSN,&MaxCompLen,
&FSFlags,FSName,100))
{
// вивід даних про диск
Label2->Caption="Ім'я тому: "+AnsiString(VolName);
Label3->Caption="Серійний номер:"+IntToHex(HIWORD(VolSN),4)+'-'
+IntToHex(LOWORD(VolSN),4);
Label4->Caption="Файлова система: "+AnsiString(FSName);
// виклик функції для одержання інформації про кластери та сектори
if (GetDiskFreeSpace(S.c_str(), &SPC,&FC,&BPS,&NC))
{
// розрахунок і вивід відомостей про вільний простір диска
Label5->Caption="Вільно: "+IntToStr(SPC*FC*BPS)+ " байт ("
+IntToStr((SPC*FC*BPS)/1024/1024)+"МБ)";
Label6->Caption="Кількість байтів у секторі: "+IntToStr(FC);
Label7->Caption="Кількість секторів у кластері:"
+IntToStr(SPC);
}
}
else
{
// реакція програми у випадку неготовності диска
Label2->Caption="Диск не готовий";
Label3->Caption=""; Label4->Caption="";
}
}

```

```
Label5->Caption=""; Label6->Caption="";  
Label7->Caption="";  
}  
}
```

Для отримання інформації про диск використовується Windows API функція `GetVolumeInformation`. Ця функція повертає інформацію про ім'я тома, серійний номер, а також характеристики його файлової системи і має наступні параметри:

- змінна `S` – ім'я диску, інформацію про який необхідно отримати;
- `VolName` – буфер, в який буде розміщено ім'я тома диска (метка диска);
- 255 – розмір буферу для імені тома диска;
- `&VolSN` – адреса змінної, в яку буде записано серійний номер;
- `&MaxCompLen` – адреса змінної, в яку буде записано максимальне значення шляху, що підтримується файловою системою диска;
- `&FSFlags` – адреса змінної, в якій знаходяться прапорці файлової системи. Може бути присутньою будь-яка комбінація прапорців, що дозволяє визначити деякі властивості файлової системи (чутливість до регістру імен файлів, підтримка імен в UNICODE і т.п.);
- `FSName` – буфер, в який буде розміщено ім'я файлової системи;
- 100 – розмір буферу для імені файлової системи.

Після виведення отриманої інформації викликається ще одна функція – `GetDiskFreeSpace`. Вона повертає інформацію про кількість кластерів і секторів, а також їх розміри для вказаного диска. Функція має наступні параметри:

- змінна `s` – ім'я диску, інформацію про який необхідно отримати;
- `&SPC` – адреса змінної, в яку буде записано кількість секторів у кластері;
- `&FC` – адреса змінної, в яку буде записано кількість байтів у кластері;
- `&BPS` – адреса змінної, в яку буде записано кількість вільних кластерів;
- `&NC` – адреса змінної, в яку буде записано загальну кількість кластерів.

На основі отриманої інформації можна провести розрахунок вільного простору на диску. Робоче вікно програми наведено на рис. 8.4.

В робочій програмі можна обирати пристрій зі списку доступних пристроїв і отримувати інформацію про параметри.

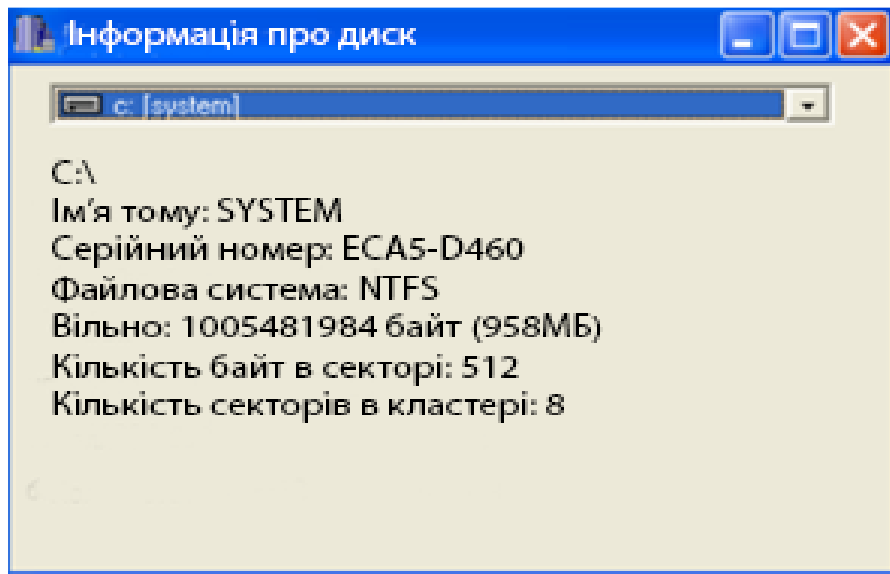


Рисунок 8.4 – Вікно робочого проекту з інформацією про диск

### 1.5 Визначення частоти процесора в режимі реального часу

Розглянемо приклад програми, здатної визначати частоту роботи процесора. Для цього створимо в C++ Builder новий проект. На формі встановимо два компонента Label і дві кнопки Button. Значення властивостей компонентів наведені в табл. 8.2, а вікно робочого проекту – на рис. 8.5.

Таблиця 8.2

Ім'я компонента	Властивість	Значення
Form1	Caption	Замір частоти процесора
Label1	Caption	Частота процесора
Label2	Caption	?MHz
Button1	Caption	Старт
Button2	Caption	Стоп
	Enabled	false

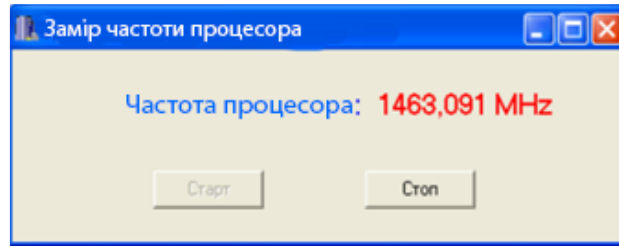


Рисунок 8.5 – Вікно програми виміру частоти процесора

У обробнику події натисканням кнопки Старт введемо наступний код:

```
bool Stop;// глобальна змінна керування запуском і зупинкою
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Button1->Enabled = false;
    Button2->Enabled = true;
    Stop = false;
    while (!Stop)
    {
        Label2->Caption = FormatFloat("0.000",GetCPUSpeed())+" MHz";
        Application->ProcessMessages();
    }
    Button1->Enabled = true;
    Button2->Enabled = false;
}
```

Для запуску визначення частоти процесора слід натиснути кнопку Старт, а для зупинки – кнопку Стоп, обробник події якої має такий вигляд:

```
void __fastcall TForm1 :: Button2Click (TObject * Sender)
{ Stop = true; }
```

Цикл while, в Button1Click, виконується до тих пір, поки глобальна змінна Stop не прийме значення true. У середині даного циклу розташовані два рядки операторів. У першій викликається функція GetCPUSpeed, за допомогою якої визначається частота процесора. У другому рядку викликається метод ProcessMessages, який дозволяє перехоплювати події, зокрема натискання кнопки СТОП.

Програмний код функції GetCPUSpeed:

```
double GetCPUSpeed()
{
    const DelayTime = 500; // час виміру в мс
    DWORD TimerHi, TimerLo;
    int PriorityClass, Priority;
    // одержання вихідних значень пріоритетів
    PriorityClass = GetPriorityClass(GetCurrentProcess);
    Priority = GetThreadPriority(GetCurrentThread);
    // зміна значень пріоритетів
    SetPriorityClass(GetCurrentProcess, REALTIME_PRIORITY_CLASS);
    SetThreadPriority(GetCurrentThread,
        THREAD_PRIORITY_TIME_CRITICAL);
    Sleep(10);
    asm
    {
        dw 310Fh
        mov TimerLo, eax
        mov TimerHi, edx
    }
    Sleep(DelayTime);
    asm
    {
```

```

dw 310Fh
sub eax, TimerLo
sbb edx, TimerHi
mov TimerLo, eax
mov TimerHi, edx
}
// установка пріоритетів у вихідне значення
SetThreadPriority(GetCurrentThread, Priority);
SetPriorityClass(GetCurrentProcess, PriorityClass);
return TimerLo / (1000.0 * DelayTime);
}

```

На самому початку записуються поточні пріоритети потоків за допомогою функцій `GetPriorityClass` і `GetThreadPriority`. Потім значення пріоритетів змінюються на максимальні за допомогою функцій `SetPriorityClass` і `SetThreadPriority`. Для одного з них встановлюємо пріоритет реального часу – `REALTIME_PRIORITY_CLASS`, а для іншого – критичний до часу пріоритет `THREAD_PRIORITY_TIME_CRITICAL`. Ці дії необхідні для отримання нашою програмою всіх ресурсів комп'ютера.

Після установки пріоритетів виконується затримка в 10 мс за допомогою виклику функції `Sleep`, з тим, щоб змінені пріоритети вступили в силу. Потім починається сам процес вимірювання частоти, який виконується за допомогою двох вставок асемблерного коду, між якими відбувається затримка на період, зазначений у константі `DelayTime`.

Після виміру роботи таймера значення пріоритету класу і потоку відновлюються у вихідні значення. Це необхідно в зв'язку з тим, що критичний до часу пріоритет і пріоритет реального часу віддають програмі всі ресурси, а за замовчуванням на такому пріоритеті працює тільки ядро ОС Windows і деякі особливо критичні додатки. Якщо програма буде постійно працювати з такими високими пріоритетами, то вона може конфліктувати з ядром і викликати збій у

роботі ОС. Саме тому бажано отримувати всі ресурси ОС Windows тільки на короткий проміжок часу.

У процесі роботи програми при виконанні вимірювань може спостерігатися деяка зміна частоти. Дане явище пояснюється тим, що багато процесорів можуть самі регулювати свою частоту в залежності від температури та інших параметрів.

## 2. Індивідуальні завдання

Розробити програмне забезпечення, яке виконує аналіз і діагностику ПК згідно з табл. 8.3.

Таблиця 8.3

№ варіанта	Завдання
1–5	Визначення параметрів BIOS (дата і версія) Визначення параметрів відеосистеми
6–10	Визначення параметрів відеосистеми Визначення частоти процесора
11–15	Отримання відомостей про диск Визначення параметрів BIOS (дата і версія)
15–20	Отримання відомостей про диск Визначення параметрів відеосистеми

## 3. Зміст звіту

1. Тема та мета роботи.
2. Результати виконання завдань. Навести алгоритм та код розробленої програми.
3. Висновки.

### Контрольні запитання

1. Де в ОС Windows зберігаються відомості про систему?
2. Як називається набір функцій Windows, за допомогою яких можна отримати відомості про компоненти комп'ютерної системи?
3. Параметри функції GetVolumeInformation та її призначення.
4. Призначення функції GetDiskFreeSpace та її параметри.

## Список літератури

1. Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Контроль та діагностика комп'ютерних систем» для студентів денної та заочної форм навчання за спеціальністю «Комп'ютерна інженерія» / Носков В.І., Панченко В.І., Гейко Г.В., Баленко О.І. – Харків: НТУ «ХПІ». – 2019. – 56 с.

2. Методичні вказівки до проведення самостійних робіт з навчальної дисципліни «Контроль та діагностика комп'ютерних систем» для студентів денної та заочної форм навчання за спеціальністю «Комп'ютерна інженерія» / Баленко О.І., Панченко В.І. – Харків: НТУ «ХПІ». – 2019. – 18 с.

3. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: посібник / В.М. Локазюк, Ю.Г. Савченко. – Київ: Видавничий центр «Академія», 2004. – 376 с.

4. Колибін Ю.М. Контроль та діагностика комп'ютерних систем. навчальний посібник / Ю.М. Колибін, Ф.А. Домнін. – Харків: ХВУ, 2000. – 274 с.

5. Гавриленко В.В. Основи надійності комп'ютеризованих систем. Навчальний посібник / В.В. Гавриленко, Р.А. Серебряков. – К.: НТУ, 2018. – 214 с.

6. Заміховський Л.М. Основи теорії надійності і технічної діагностики систем: навчальний посібник./ Л.М. Заміховський, В.П. Калявін. – Івано-Франківськ: Вид-во «Полум'я», 2019. – 360 с.

7. Tanenbaum A.S. Computer networks: 5th ed. / A.S. Tanenbaum, D.J. Wetherall. – Boston: Prentice Hall, 2011. – 962 p.

8. Тарасенко В.П. Надійність комп'ютерних систем: навчальний посібник / В.П. Тарасенко, А.Ю. Маламан, Ю.П. Черниченко, В.І. Корнійчук. – Київ: «Корнійчук», 2007. – 256 с.

9. Черв'яков В.Д. Основи надійності об'єктів системотехніки: навчальний посібник / В.Д. Черв'яков, А.В. Павлов, О.Ю. Журавльов. – Суми: Сумський державний університет, 2011. – 245 с.

## ЗМІСТ

<b>Вступ</b> .....	3
<b>Лабораторна робота 1.</b> Програми діагностики ПК.....	4
<b>Лабораторна робота 2.</b> Програми тестування жорстких дисків.....	16
<b>Лабораторна робота 3.</b> Тестування ПК програмою «AIDA64».....	26
<b>Лабораторна робота 4.</b> Тестування ПК програмою «Sisoftware Sandra».....	30
<b>Лабораторна робота 5.</b> Визначення кількісних характеристик надійності за статистичними даними про відмови елементів.....	35
<b>Лабораторна робота 6.</b> Завадостійке кодування. Коди Хемінга.....	40
<b>Лабораторна робота 7.</b> Діагностика мереж WINDOWS вбудованими програмами.....	46
<b>Лабораторна робота 8.</b> Розробка програм для діагностики параметрів ПК.....	51
<b>Список літератури</b> .....	65

Навчальне видання

Методичні вказівки до виконання лабораторних робіт  
з навчальної дисципліни «Контроль та діагностика комп'ютерних систем»  
для студентів денної та заочної форми навчання  
за спеціальністю «Комп'ютерна інженерія»

2-ге видання, перероблене та доповнене

Укладачі:

НОСКОВ Валентин Іванович

ПАНЧЕНКО Володимир Іванович

БАЛЕНКО Олексій Іванович

Відповідальний за випуск проф. Олександр ЗАКОВОРОТНИЙ  
Роботу до видання рекомендував проф. Микола ЗАПОЛОВСЬКИЙ

В авторській редакції

План 2024 р., поз. 557  
Підп. до друку \_\_\_\_. Формат 60x84 1/16.  
Папір офсет. Друк ризографічний. Ум. друк. арк. 2.

---

Видавничий центр НТУ «ХПІ»,  
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.  
61002, м. Харків, вул. Кирпичова, 2

---

Електронна версія