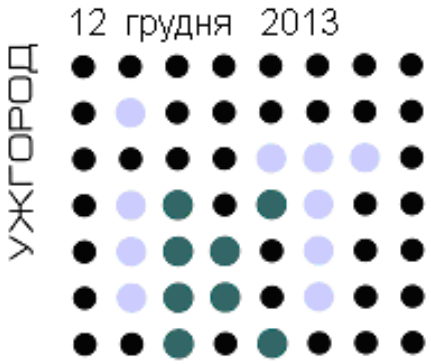
КАФЕДРА ТВЕРДОТІЛЬНОЇ ЕЛЕКТРОНІКИ З/С
ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ДВНЗ „УжНУ”

ЗАКАРПАТСЬКЕ ФІЗИЧНЕ ТОВАРИСТВО

АКАДЕМІЯ ТЕХНОЛОГІЧНИХ НАУК УКРАЇНИ



II НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ

"ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У ЖИТТІ СТУДЕНТІВ ТА МОЛОДИХ НАУКОВЦІВ ЗАКАРПАТТЯ"

МАТЕРІАЛИ КОНФЕРЕНЦІЇ

УЖГОРОД - 2014

Матеріали ІІ Науково-практичної конференції «Інформаційні технології у житті студентів та молодих науковців Закарпаття». – Ужгород: ДВНЗ «УжНУ», 2014. – 48 с.

Представлені результати теоретичних та експериментальних досліджень в таких напрямках: фундаментальні та прикладні дослідження в області інформаційно-комунікаційних технологій; нові наукові і практичні результати застосування сучасних інформаційно-комунікаційних технологій; ІТ-інновації для науки та освіти; мікропроцесорні інформаційні системи; методи і засоби обробки сигналів і зображень; веб-технології; комп'ютерні системи та мережі; високопродуктивні обчислювальні системи; паралельні та розподілені обчислення; захист інтелектуальної власності; нові форми навчання з використанням інформаційно-комунікаційних технологій; інформаційне право.

Матеріали підготовано до друку організаційним комітетом конференції і подано в авторській редакції.

Рекомендовано до друку Вченою Радою Ужгородського національного університету, протокол № 7-2013/2014 від 20.03.2014 р.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Морозов А.О. - Почесний голова, Президент Академії технологічних наук України;
Ващук Ф.Г. - голова, д.т.н., професор, ректор ДВНЗ „УжНУ”;
Різак В.М. - заступник голови, д. ф.-м. н., професор, завідувач кафедри ТЕІБ;
Головко В.П. - к. ф.-м. н., доцент кафедри ТЕІБ;
Юркович Н.В. - к. ф.-м. н., доцент кафедри ТЕІБ;
Чобаль О.І. - ст. викладач кафедри ТЕІБ;
Кокряшкіна Я.І. - секретар, аспірант.

ЖУРІ

Морозов А.О. - Почесний голова, Президент Академії технологічних наук України;
Ващук Ф.Г. - голова, д. т. н., професор, ректор ДВНЗ „УжНУ”;
Різак В.М. - заступник голови, д. ф.-м. н., професор, завідувач кафедри ТЕІБ;
Король І.І. - д. ф.-м. н., професор математичного факультету ДВНЗ „УжНУ”;
Повхан І.Ф. - к. т. н., в. о. декана факультету інформаційних технологій ДВНЗ „УжНУ”;
Головко В.П. - к. ф.-м. н., доцент кафедри ТЕІБ;
Дробнич О.В. - к. ф.-м. н., доцент факультету інформаційних технологій ДВНЗ „УжНУ”;
Юркович Н.В. - к. ф.-м. н., доцент кафедри ТЕІБ;
Чобаль О.І. - ст. викладач кафедри ТЕІБ;
Скубенич М.М. - викладач кафедри ТЕІБ;
Маркевич П.В. - аспірант;
Кокряшкіна Я.І. - аспірант;
Данилюк П. - студент;
Шварц О. - студент.

ЗМІСТ

ПРОЕКТУВАННЯ ДВІЙКОВОГО ПАРАЛЕЛЬНОГО СУМАТОРА З ПАРАЛЕЛЬНИМ ПЕРЕНОСОМ

І.Ю. Король, Н.В. Ракущинець

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Університетська, 14

Вступ

Зростання обсягів інформації зумовлює необхідність збільшення ефективності використання обчислювальних пристроїв цифрової обробки інформації, яка залежить від методу формування, перетворення, обробки, схемотехнічної реалізації, форми подання інформації та, зокрема, від швидкості виконання арифметичних операцій при реалізації перетворень. Суть обробки інформації у цифровій формі полягає у виконанні заданої послідовності найпростіших арифметичних і логічних операцій над числами.

Для подання чисел в цифрових системах найчастіше використовується двійкова система числення. Проте, час виконання арифметичних операцій в двійковій системі залежить від розрядності суматора, основного функційного компонента арифметико-логічного пристрою, зокрема – внаслідок формування та поширення міжрозрядних переносів.

Основна частина

Принцип додавання багаторозрядних двійкових чисел полягає у тому, що у кожному з розрядів виконуються однотипні дії: визначається цифра суми шляхом додавання по модулю 2 цифр доданків і переносу з попереднього розряду і формується перенос, який надходить до наступного розряду. Ці дії реалізуються двійковим однорозрядним суматором. Така однотипність дій при додаванні різних розрядів багаторозрядних доданків дозволяє реалізувати багаторозрядні суматори як у послідовному вигляді – за рахунок послідовного виконання додавання розрядів за допомогою одного однорозрядного суматора, так і у паралельному – за допомогою схеми, що вміщує кілька однотипних фрагментів (за числом розрядів доданків), кожен з яких має свій однорозрядний суматор.

Явна перевага суматора послідовної дії полягає у малих апаратних витратах на його побудову, однак тривалість операції додавання пропорційна розрядності доданків. Такий суматор, у порівнянні з більш складним паралельним, має меншу швидкодію.

Паралельний багаторозрядний суматор (*ripple carry adder*) складається з такої кількості однорозрядних суматорів, яка дорівнює кількості розрядів чисел, які додаються. Усі розряди доданків повинні одночасно (паралельно) надходити до пристрою додавання. Сигнал переносу передається від розряду до розряду послідовно, утворюючи на виході значення старшого розряду суми. Схема паралельного чотирирозрядного комбінаційного суматора з послідовним переносом показана на рис. 1.

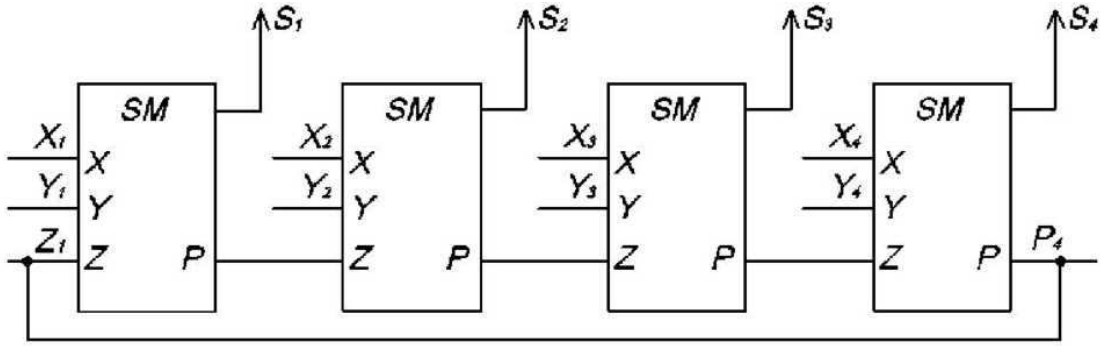


Рис. 1 Функціональна схема чотирирозрядного паралельного суматора з послідовним переносом

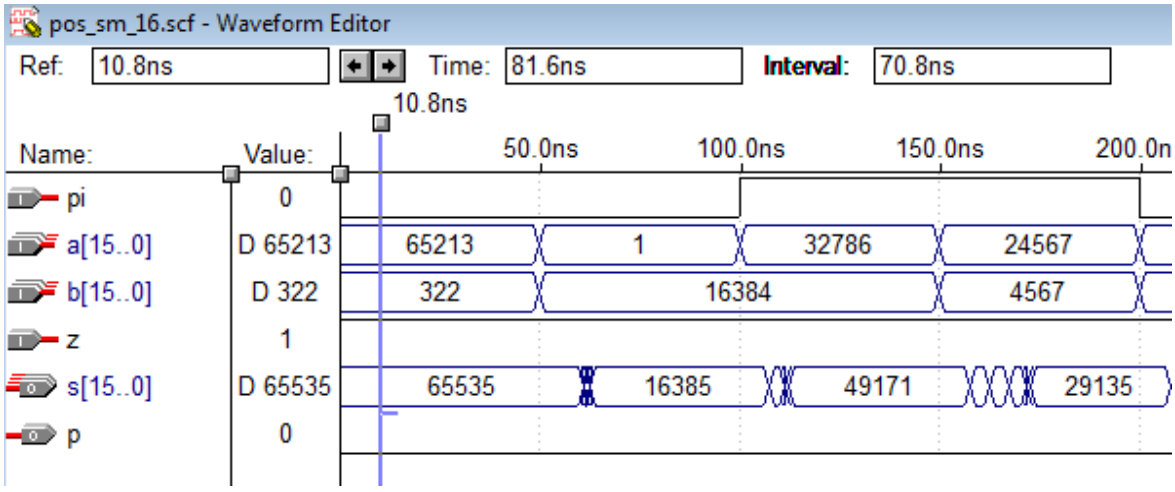


Рис. 2 Моделювання роботи 16-розрядного паралельного суматора з послідовним переносом

Недоліком паралельного суматору з послідовним переносом є те, що отримання результату у старшому розряді суматора можливе

тільки після завершення розповсюдження переносу по усіх розрядах, що знижує швидкодію пристрою. Саме тому у схемах паралельних суматорів організовують паралельний перенос.

Суматори з паралельним переносом призначені для забезпечення найбільшої швидкодії виконання арифметичних операцій. Гранична швидкодія відрізняється в 2, 3 рази від елементарної затримки. Проте в реальних схемах таких границь досягнути неможливо, оскільки побудова суматорів багаторозрядних чисел на основі нормальних форм привела б до громіздких схем. Реальні схеми мають модульну структуру, тобто складаються з підсхем (розрядних схем), що значно спрощує їх, проте не дозволяє отримати гранично можливу швидкодію.

Суматори з паралельним переносом (*carry lookahead adder*) не мають послідовного розповсюдження переносу по усіх розрядах розрядної сітки. У кожному розряді результати формуються одночасно. Для цього у кожному однорозрядному двійковому суматорі додатково формується сигнал розповсюдження переносу. Ідея прискорення переносу полягає у тому, що при $X_i=Y_i=1$ у i-му розряді буде мати місце перенос до наступного розряду, незалежно від наявності переносу із попереднього розряду. Таким чином, у цьому випадку можна передавати сигнал переносу для обчислення старших розрядів, не чекаючи закінчення формування переносу із молодших розрядів. Для реалізації прискореного переносу схему паралельного суматора дещо ускладнюють додатковими спеціальними схемами (рис. 3), на входи яких надходять сигнали, що формують сигнал переносу, тобто ті, від яких залежить його наявність або відсутність. Такими сигналами є зовнішній вхідний перенос, якщо він існує та значення всіх розрядів доданків молодших відносного даного розряду. Для опису роботи суматора з паралельним переносом вводять дві додаткові допоміжні функції:

1)D – функція генерації переносу (*carry generation*). D=1, коли доданки даного розряду є такими, що перенос C_i у сусідній старший розряд дорівнює 1 незалежно від значення вхідного переносу C_{i-1} даного розряду, тобто D=1, якщо в даному розряді суматора генерується перенос в старший розряд, то $D=a_i \& b_i$.

2)F – функція прозорості або розповсюдження (*carry propagation*). F=1, якщо доданки даного розряду є такими, що вхідний перенос у даний розряд C_{i-1} дорівнює 1, то перенос C_i у сусідній

старший розряд також буде 1, тобто F=1, якщо тракт переносу даного розряду є прозорим для вхідного сигналу переносу C_{i-1}, то $F= a_i \# b_i$.

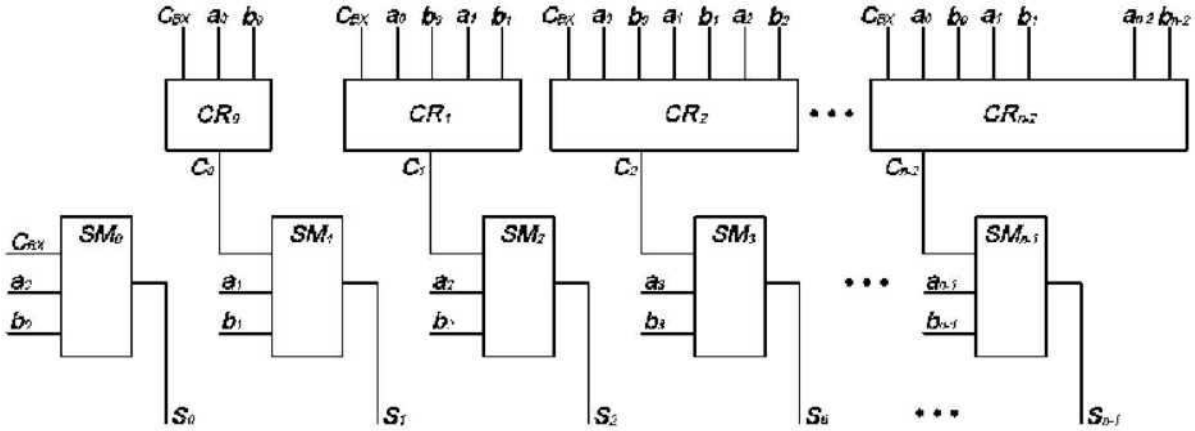


Рис. 3 Паралельний багаторозрядний суматор з паралельним переносом, де: CR_i — блок логічних схем формування сигналу переносу для кожного i-го суматора

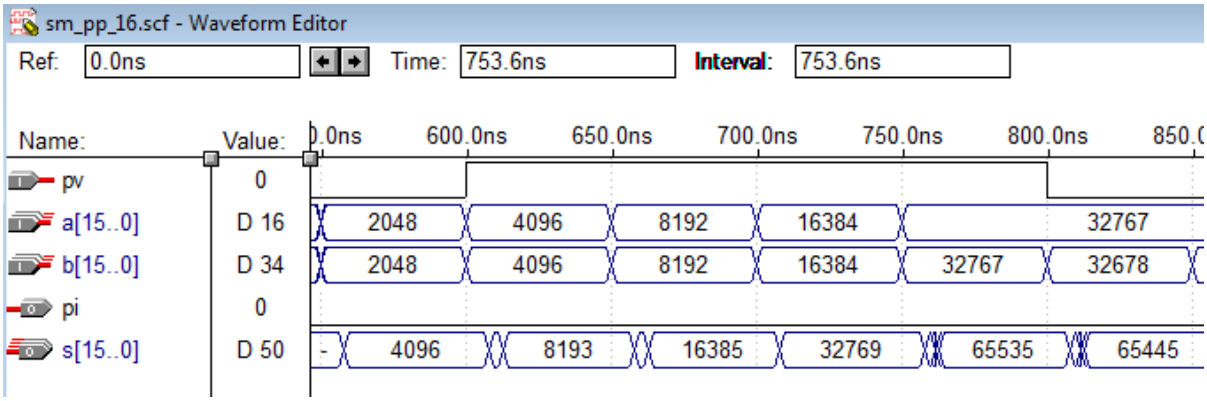


Рис. 4 Моделювання роботи 16-розрядного паралельного суматора з паралельним переносом

Тривалість додавання, судячи з наведеної схеми суматора, не залежить від його розрядності, що є характерною ознакою структур з паралельним переносом взагалі, не тільки суматорів. Проте, фактично це є не зовсім так, оскільки із збільшенням розрядності суматора підвищується навантаження елементів схеми, що збільшує їх затримки. Зокрема, коефіцієнт розгалуження елементів, що реалізують функції прозорості, дорівнює $n^2/4$, тобто квадратично залежить від розрядності суматора. Тому збільшення розрядності уповільнює процес додавання. Якщо розрядність до $n=4$, то перевагу у часі додавання мають переважно більш прості суматори з послідовним переносом, після $n=8$ з'являються перевантажені

елементи і елементи з великою кількістю входів, що уповільнює роботу суматора та потребує для спрощення схеми введення додаткових елементів зі своїми затримками.

Таблиця 1. Залежність затримки передачі даних від розрядності схеми

Розрядність,біт	Паралельний суматор з паралельним переносом	Паралельний суматор з послідовним переносом
4	9,6 *ns*	9,6 *ns*
8	18,3 *ns*	27,5 *ns*
16	45 *ns*	55,5 *ns*
32	85,2 *ns*	116,5 *ns*
64	198,5 *ns*	233,5 *ns*
128	494,5 *ns*	714 *ns*
256	1423,5 *ns*	1917,6 *ns*

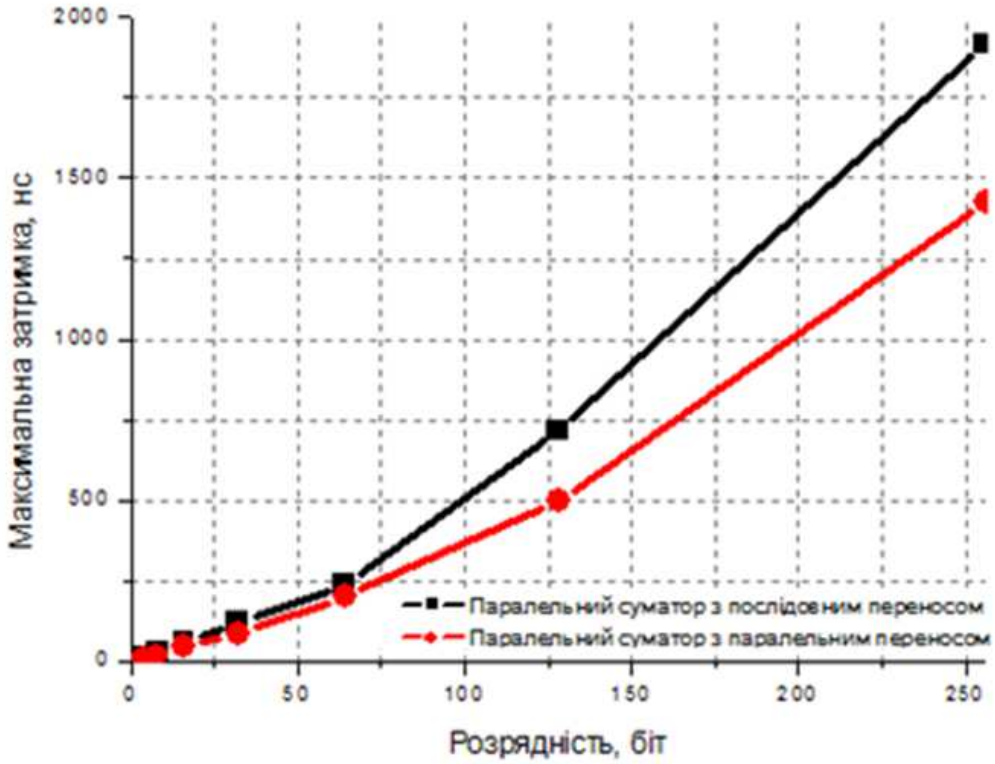


Рис. 5 Графік залежності затримки передачі даних від розрядності схеми.

Висновки

Підвищення швидкодії схем забезпечується за рахунок принципу паралельності – організації одночасного надходження усіх розрядів операндів на пристрій додавання та формування переносів, необхідних для виконання арифметичних операцій. Основною і визначальною ознакою затримки будь-якої із схем виступає час, необхідний для формування міжрозрядних переносів, що безпосередньо залежить від методів та засобів організації переносів. Саме тому актуальною задачею на сьогодні виявляється дослідження класу суматорів, які б не містили довгих ланок переносу.

На основі аналізу схем багаторозрядних суматорів, в основу яких покладено максимальну затримку надходження даних на вихід, встановлено, що відмінність у швидкодії спостерігається при побудові схем розрядністю в 16 бітів і більше.

Таким чином, проаналізувавши схеми організації міжрозрядних переносів при виконанні арифметичних операцій у багаторозрядних суматорах, слід зазначити, що кожна із структур має свої певні недоліки – послідовні суматори мають досить великий час затримки розповсюдження сигналу переносу, хоча і відносно просту апаратну реалізацію, паралельні – досить складну апаратну реалізацію, яка виправдовується лише при операціях з багаторозрядними числами.

Література

1. Аряшев С.И. Исследование методов построения многорозрядных быстродействующих сумматоров / С.И. Аряшев, П.С. Зубковский // http://www.library.mephi.ru/data/scientificsessions – 154.html. – 2003.

2. Угрюмов Е.П. Цифровая схемотехника / Е.П. Угрюмов. – СПб.: БХВ-Петербург, 2004. – 528 с.

3. Король І.Ю. Мова опису апаратури AHDL: Навчальний посібник для студентів напрямку 6.050102 – «Комп’ютерна інженерія»/ І.Ю. Король–Ужгород: УжНУ, 2008. –104 с.

4. Бабич М.П. Комп’ютерна схемотехніка: Навчальний посібник / М.П. Бабич, І.А. Жуков. – К.: НАУ, 2002. – 508 с.

ПРОГРАМНІ ЗАСОБИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ВІДЕОДАНИХ

Тиводар А.М.

ДВНЗ „Ужгородський національний університет"
88000, Ужгород, вул. Університетська, 14

Актуальність розробки засобів розпізнавання об'єктів у потоці відеоданих зумовлена збільшенням обсягів відеоінформації, яка потребує спеціальних методів аналізу. Такі розробки застосовуються у наступних сферах:

– технічна діагностика: на виробництві часто виникає проблема автоматизувати контроль якості деталей. Задача полягає у тому, щоб виявити, чи є деталь дефектною, чи ні;

– медична діагностика – найтиповіша ситуація полягає в тому, що ті чи інші захворювання діагностуються на основі аналізу кардіограм, рентгенівських знімків і т.п.;

– розпізнавання літер – для того, щоб створити текстовий документ, з яким може працювати текстовий редактор, необхідно впізнати на зображенні окремі літери;

– охоронні системи – потрібно ідентифікувати деяку особу, щоб визначити, чи має вона право входити на територію, що охороняється.

Також можна назвати вимірювання інтенсивності дорожнього руху, визначення числа зайнятих і вільних місць для паркування і т.д.

Однією з найважчих задач обробки відеозображення є проблема виділення і розпізнавання рухомих об'єктів при наявності різного роду перешкод і створення на цій основі системи моніторингу. Головна задача таких систем – інформувати людину про ситуацію, що склалася в полі зору камери, і по можливості приймати певні рішення.

Розпізнавання об'єктів – це віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних. Задача ідентифікації полягає у тому, щоб вирізнити певний конкретний об'єкт серед йому подібних.

Умовно обробку відеозображення можна розділити на наступні етапи:

– відділення переднього плану – проводиться відділення рухомих фрагментів зображення від нерухомих, або таких, що належать задньому плану. Від того, на скільки правильно вирішена ця задача, залежать всі наступні етапи. Завдання виділення фону є досить непростим, оскільки на зображеннях, які утворюються в реальній відеозйомці, відсутні абсолютно незмінні фрагменти. Це пов'язано як з властивостями реальних камер, які мають досить помітні власні шумами, так і з трактом передачі відеоінформації від камери до системи обробки, в якому зображення перед пересиланням часто ущільнюється, а потім декодується, що може призвести до додаткових спотворень сигналу. Більше того, на сцені часто присутні об'єкти, які хоч і є в нашому розумінні нерухомими, проте володіють певними динамічними характеристиками (наприклад, дерева, що коливаються під впливом вітру). Облік перерахованих факторів при виділенні фону призводить до необхідності розробки досить гнучкої системи, яка формує модель заднього плану;

– виділення й класифікація рухомих об’єктів – відеопотік може містити різноманітні рухомі об'єкти. Одні з них представляють інтерес для моніторингу – наприклад, зображення людей, транспортних засобів, тварин. Інші об'єкти, такі як гілки дерев, тіні, обертові двері, створюють перешкоди при відеоспостереженні. Вся сукупність рухомих об'єктів складає передній план, причому апріорно невідома кількість таких об'єктів, їхній розмір і їх взаємне розсташування. Для того щоб провести класифікацію об'єктів, що рухаються, потрібно спочатку відокремити кожний об'єкт від інших. Ця операція називається сегментацією об'єктів. Результатом проведення сегментації є або позначений набір пікселів переднього плану, або вказання вершин мінімально можливого прямокутника, всередині якого знаходиться виділений об'єкт;

– відслідковування траєкторії руху знайдених об’єктів – для цього потрібно встановити відповідність між знайденими об’єктами на послідовних кадрах. Це дає змогу отримати інформацію про траєкторію, швидкість і напрям руху;

– розпізнавання й опис рухів об’єктів – в ідеальному випадку система повинна видавати повідомлення типу: автомобіль виїхав з парковки і під’їхав до воріт.

Сукупність нерухомих об’єктів називають фоном або заднім планом, а рухомі об’єкти – переднім планом.

Розрізняють наступні методи відділення переднього плану:

– методи вирахування фону – полягає в попіксельному порівнянні поточного кадру з шаблоном, який зазвичай називають моделлю фону. Ця модель, яка представляє собою опис сцени без рухомих об’єктів, повинна постійно обновлюватися. Дані методи навіть в простішій реалізації мають високі вимоги до ресурсів системи;

– ймовірнісні методи – це такі, де зміна значень пікселів в часі розглядаються як попіксельний процес, тобто часовий ряд, який складається з скалярних величин і векторів. У результаті фон представляє собою Гауссову суміш. Враховуючи час життя і дисперсії кожного гауссіана, можна визначити, які з них відносяться до фону;

– методи часової відмінності – вони дозволяють відділяти передній план від фону за допомогою операції попіксельного порівняння двох і більше послідовних кадрів;

– методи оптичного потоку – для відеофрагменту, який містить деякі рухомі об’єкти, можна обрахувати напрямок й швидкість руху в кожній точці кадру. Інформація про оптичний потік використовуються для просторової сегментації зображення. За допомогою цього методу можна провести найбільш точне виділення рухомих об’єктів, але алгоритм потребує значних ресурсів, і дуже залежить від шумів камери.

Методи з різних груп мають різну складність реалізації і, відповідно, відрізняються необхідними вимогами до обчислювальних ресурсів. У більшості випадків завдання обробки відеопотоку здійснюється в реальному часі, і часто на одному комп'ютері обробляється інформація, що отримується одночасно від декількох камер. При розробці подібної системи розробники віддадуть перевагу більш простим методам порівняння з базовим фоном втрачаючи якість, ніж методам оптичного потоку, який вимагає досить значних ресурсів. У той же час при обробці відеоархівів, (наприклад. для їх індексування), можна застосувати більш складні методи, що дозволяють досягти кращої якості обробки.

Приклад побудови моделі заднього плану різними методами зображено на рис. 1.

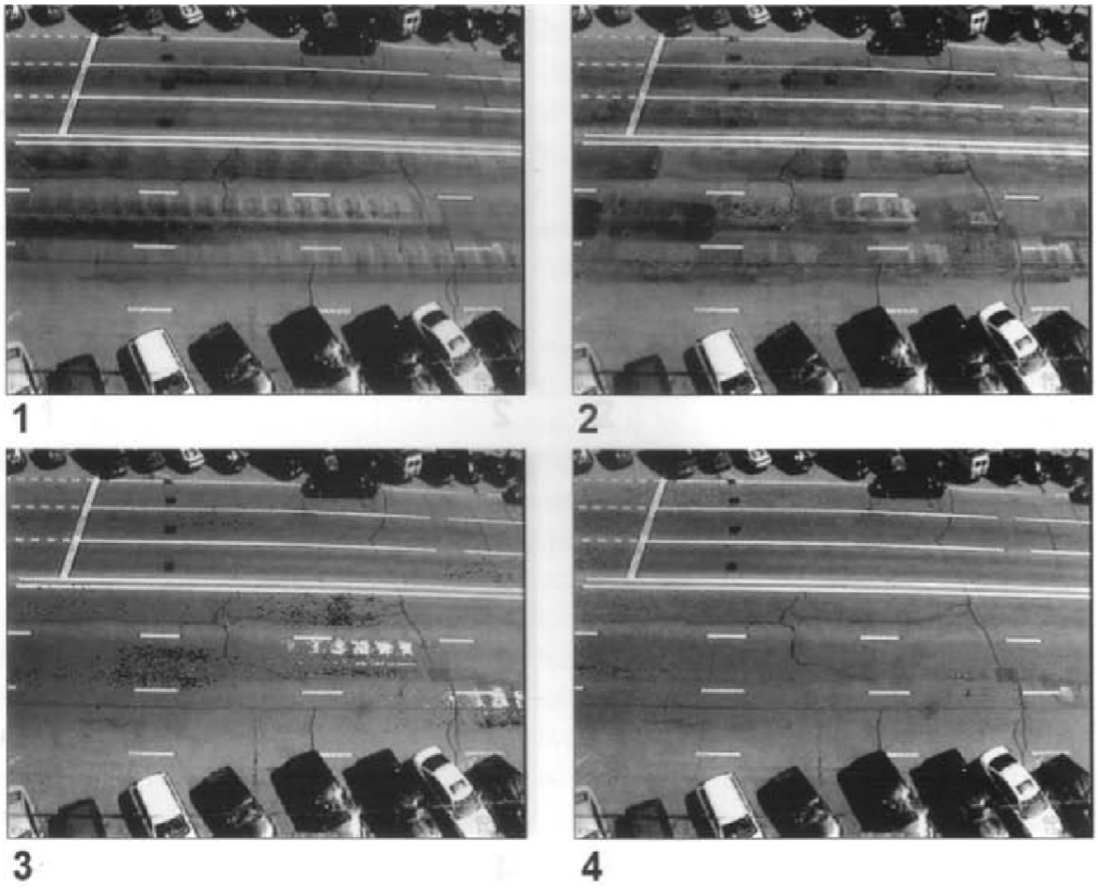


Рис.1 Приклад побудови моделі заднього плану (фону): 1 – методи порівняння з базовим фоном, 2 – методи часової відмінності, 3 – імовірнісні методи, 4 – методи, які базуються на нейромережах.

Для виокремлення переднього плану використовується метод часової відмінності. Методи часової відмінності відокремлюють передній план від фону за допомогою попіксельного вирахування двох або більше послідовних кадрів. Очевидно, що методи часової відмінності добре визначають динамічні зміни сцени, але зазвичай не можуть виділити цілком всі однорідні пікселі одного об'єкта. що призводить до фрагментованості виділених об'єктів (часто всередині них утворюються порожнеча). Крім того, цими методами вдається виявляти об'єкти, що зупинилися. Тому найчастіше методи тимчасової різниці використовуються разом з іншими методами (наприклад, ймовірнісними чи порівнянням з базовим фоном), що дозволяє їм досить стійко відокремлювати передній план від фону за невеликий час.

Існує кілька варіантів методів часової відмінності. Найпростіший з них полягає в наступному. Для кожного пікселя

поточного кадру складається його різниця з відповідним пікселем попереднього кадру:

$$\Delta(x,y,t) = |I(x,y,t) - I(x,y,t-1)| \quad (1)$$

Якщо ця величина перевищує порогове значення δ, то даний піксель вважається передньоплановим, в іншому випадку – належить до фону. Трудність викликає правильний вибір порогового значення. Для його вибору можна використовувати такий алгоритм:

Обчислимо для кожного пікселя спочатку усереднене значення величини $\Delta(x, y, t)$ по області зображення Ω навколо розглянутого пікселя:

$$\Delta_l(x,y,t) = \frac{1}{N}\sum_{x,y \in \Omega} \Delta(x,y,t) \quad (2)$$

де N - кількість пікселів в Ω. Потім визначимо різницю між (1) й (2)

$$\Delta_c(x,y,t) = \Delta_l(x,y,t) - \Delta_l(x,y,t) \quad (3)$$

Знайдемо число пікселів, що лежать в тих областях зображення, які вважаються фоновими:

$$M = \{(x,y): (|\Delta_c(x,y,t)| < 2\,\sigma_\Delta) \cap (|\Delta_l(x,y,t)|) < m_\Delta\} \quad (4)$$

де σ_Δ- стандартне відхилення для шуму камери, розраховане для різниці кадрів, а m_Δ позначає медіанне значення $\Delta_l(x, y, t)$. Тоді граничне значення δ визначається як:

$$\delta = \frac{1}{|M|}\sum_{(x,y)} \Delta_i(x,y,t). \quad (1.16)$$

де $|M|$ - число елементів множини M. До цього значення порога може бути також додано доданок, що представляє собою граничне значення для шуму камери. Більш точні результати можна отримати, якщо брати до уваги динаміку сцени, зафіксовану на більш ніж двох послідовних кадрах.

Для тестування алгоритмів реалізовано програмний засіб, який дозволяє виокремлювати рухомі автомобілі у відеопотоці. Також виконується підрахунок кількості автомобілів з можливістю зберігання зображення кожного з них.

Проблеми розпізнавання легко вирішуються людьми, причому робиться це, як правило, підсвідомо. Спроби ж побудувати штучні системи розпізнавання не настільки переконливі. Основна проблема полягає у тому, що часто неможливо адекватно визначити ознаки, на основі яких слід здійснювати розпізнавання. Для задач, яким такі ознаки вдається виділити, штучні системи розпізнавання набули значного поширення і широко використовуються.

Література

1. Лукьяница А.А., Шишкин А.Г. Цифровая обработка видеоизображений. – М., 2009. – 421 с.
2. Лукьяница А.А., Шишкин А.Г. Пробемы обработки видеоизображений в системах мониторинга. – М., 2007. – 621 с.
3. Методы компьютерной обработки изображений. / Под ред. В.А. Сойфера. – М.: Физматгиз, 2001. – 784 с.
4. Рыбаков О.С. Об оценке параметров контура на изображении. Материалы 2-й Международной научно-практической конференции: Теория, методы и средства измерений, контроля и диагностики. – Новочеркасск: НПИ, 2001 с.13.

ЕКСПЕРИМЕНТАЛЬНІ ПРОТОКОЛИ ПІДКЛЮЧЕННЯ GOOGLE В ХМАРНИХ ОБЧИСЛЕННЯХ

Шварц О.Р.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Волошина, 54

На сьогодні, IT інфраструктура і в часності її веб частина мають тенденцію надзвичайно швидкого розвитку і в результаті цього збільшення об'єму пам'яті з якою вони працюють. Об'єм трафіка інтернету колосально збільшився: в 1991 році він становив 0,001 пБ/місяць, в 2013 році він збільшився до 30 тис. пБ/місяць.

На даний момент весь веб - 99% працює через http протокол зв’язку, який був розроблений 1991 році. Якщо розглянути на прикладі завантаження веб сторінки то починаючи з 1996 року(виходу версії HTTP1.0) і до 2013 року швидкість завантаження одної і тої самої веб сторінки виросла на 100 мс при загальному часі завантаження сторінки в 1000мс (1000мс → 900мс).

Ці факти означають тільки одне, що все розвивається і зважаючи на те, що це IT сфера все розвивається надзвичайно швидко.

По цій самій причині компанія Google в 2012 році розробила і анонсувала новий протокол передачі веб контенту SPDY. Який успішно використовує в своїх продуктах. По офіційним даним тестування даного протоколу - швидкість завантаження веб сторінки зросла майже в 2 рази(час завантаження http 4,42с → spdy 2,66с).

Але на теперішній час вливання веб продукції у використання цього типу протоку зв'язку йде дуже повільними темпами. Лише пару веб ресурсів перейшло на цей протокол. Першими звичайно були Google і всі його продукти(gmail, google drive, g+ ...) і в 2013 році до них приєднались Twitter і Facebook. На даний момент, поле користувачів SPDY становить менше 1% веб простору.

Чому ж інші веб ресурси не мігрують до SPDY?

1) Перш за все, кожній компані прийдеться потратити час та кошти на переконфігурування їхніх продуктів, що може бути затратним.

2) Так як SPDY - надзвичайно новий протокол. Перша версія аннонсована в 2012 році, а перша стабільна версія, яку рекомендовано використовувати в продакшин серверах була випущена в 2013. Зважаючи на це спеціалістів по правильному впровадженні даного протоколу в веб продукт не так і багато. Це збільшує ризик і також час та кошти при впровадженні.

3) Для всіх операцій вашого сайту з SPDY ви маєте мати SSL Certificate. В HTTP, SSL сертифікат був необов'язковим. Я думаю розробники навіть не задумували над цим, але для деяких країн не має можливості отримати SSL сертифікат (наприклад Іран). Та й в загалі я не бачу потреби використовувати сертифікат для передачі всіх без виключення даних.

Отже, трохи по принципі роботи SPDY протоколу. SPDY було придумане, щоб зменшити швидкість завантаження веб сторінки та швидкості передачі даних в веб просторі.

В основі цього лежить пару принципів:

1) Стиснення заголовків - дана система стискає заголовки запита та відповіді, шо дає можливість передавати менший розмір даних і зменшити час передачі.

2) Мультипоточність - має можливість обробляти кілька запитів одночасно і також відправляти кілька запитів одночасно. Це одна з основних речей, які дають такий приріст швидкості.

3) Пріоритизація запитів - сторона клієнта веб застосунку дає можливість виставити пріоритети запитів до сервера. Таким чином розробник має можливість завантажити для користувача результативний контент швидше чим інші дані на веб сторінці.

4) “Server push” - технологія, яка дає можливість серверу самостійно відправляти дані до клієнтської частини без запиту від клієнта. Це також одна з самих важливих частин даного протоколу, так як це дає можливість не тільки оптимізувати саму передачу даних, а ще й структуру клієнтського застосунку.

5) “Server hint” - технологія, яка повідомляє клієнтську частину про нові дані, які з'явились на сервері і які теоретично потрібні для клієнта. В цьому випадку застосунок може відправити запит на отримання цих даних. На відміну від “Server push” технології, “Server hint” не відсилає дані, а просто сповіщає про їхню появу.

Результати даної роботи:

1) Було розроблена модель роботи сервер - веб клієнт додатку з повним використанням SPDY як протоку передачі даних між сервером і клієнтом. Визначено переваги в швидкості і більшому рівні захищеності даних під час їх передачі. Для саме цього пункту я взяв за мету дослідити протокол в реальних умовах і запевнитись в нього ефективності. Так як інформації та і готових прикладів по реалізації даного типу комунікації(сервер - веб клієнт) мало. В ході роботи було зпроектовано модель комунікації сервер - веб клієнт і визначено пару значних втрат в швидкості від використання TSL(покращеного SSL).

2) Була досліджена специфіка роботи протоколу для мобільний застосунків (server - mobile client). Так як на даний момент, SPDY не підтримується Safari браузером виникла потреба розробити рішення для користувачів платформи iOS, яка була досягнута шляхом емуляції веб запитів. Також досліджено швидкість передачі даних даним протокол, що показало майже в 2 рази зменшення часу отримання та обробки запита.

3) В процесі роботи над статтею, я стикнувся з декількома проблема саме хостингу веб застосунків, які працюють через SPDY протокол. Саме тому була запропонована концепція хостингу додатків використовуючи хмарні обчислення. В основу цього було покладено PaaS модель Cloud hosting. Покишо вона представлена мною у формі прототипу. В основу були покладені такі вимоги:

- Cloud Hosting з повною підтримкою SPDY для Ruby on Rails застосунків.
- PaaS - система для простого керування серверним простором, забезпечення повної підтримки програмного забезпечення без потреби втручання зі сторони клієнта.
- Автоматичне налаштування та завантаження веб застосунків на сервер без потреби втручання з клієнтської сторони.
- Автоматична конвертація веб частини продукта на SPDY протокол передачі даних.

Наступні кроки:

1) Розробка платформи, яка б могла проводити автоматичне переведення веб застосунка для пітримання основних елементів протоколу. Також автоматичного завантаження контенту на Cloud Hosting

2) Покращення самого протоколу передачі даних:

- Опційне налаштування SSL сертифікату
- Специфіка мультипоточності даного протоколу має певні мінуси. При посиланні 1 запиту однопоточний тип передачі є більш затратним чим передача того самого 1 запиту мультипоточним способом.

3) Розробка специфічного серверу для RoR застосунків, який буде використовувати SPDY 4.0 як протокол передачі даних. На даний момент існує сервер такого типу який використовує SPDY 3.0. Даний пункт може бути реалізований з використанням попередньої версії сервера як основи.

На даний момент немає готового PaaS Cloud Hosting який би підтримував SPDY, тому дана концепція є актуальною.

Отже, при розробці додатків керуючись цими рішеннями кожен розробник може досягнути 55% приросту до швидкості обміну даними в його проекті.

На сьогодні менше 1% веб простору використовує даний протокол передачі даних, тому ваш додаток має реальну можливість виграшу у швидкості комунікації веб даних.

ПРОЕКТУВАННЯ ДВІЙКОВО-ДЕСЯТКОВОГО СУМАТОРА

Король І.Ю., Панько В.В.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Університетська, 14

Вступ

Постійне розширення галузей застосування засобів цифрової обчислювальної техніки веде до необхідності вирішення більш складних задач. Це, в свою чергу, визначає постійне зростання вимог до продуктивності обчислювальних систем та підвищення їх швидкодії.

Для подання чисел в цифрових системах часто використовується двійково-десяткова система числення. Оскільки при цьому вилучаються витрати часу на переведення чисел з десяткової системи числення в двійкову і навпаки.

Двійково-десяткові коди є комбінацією двійкового й десяткового кодів. При двійково-десятковому кодуванні зберігається розташування десяткових розрядів, але кожен з них (цифри від 0 до 9) подається двійковим кодом – комбінацією з чотирьох двійкових символів («0» і «1»), що називають тетрадою. Кількість різних комбінацій, які можуть бути утворені з чотирьох символів, дорівнює 16. Водночас, для зображення розрядних коефіцієнтів (цифр) десяткової системи потрібно лише десять комбінацій, а останні шість можливих комбінацій зайві. Оскільки для кодування можуть бути використані будь-які 10 із 16 комбінацій, то це призводить до багатозначності розв’язання задачі двійково-десяткового кодування.

Основна частина

Метою даної статті є ознайомлення з принципами роботи двійково-десяткових суматорів, їх структурою, класифікацією, різними методами реалізації. Ця тема є досить актуальною у зв’язку з тим, що людина все своє життя використовує десяткову систему числення, а комп’ютер – двійкову.

Отже, розглянемо детально двійково-десяткові суматори та принципи додавання багаторозрядних чисел. Цей принцип полягає у тому, що у кожному з розрядів виконуються однотипні дії:

визначається сума шляхом додавання цифр тетрад і переносу з попереднього розряду. Якщо сума перевищує число дев'ять, то здійснюєтьсь корекція (додавання числа 0110_2) і формується перенос, який надходить до наступного розряду. Ці дії реалізуються двійково-десятковим однорозрядним суматором (рис. 1).

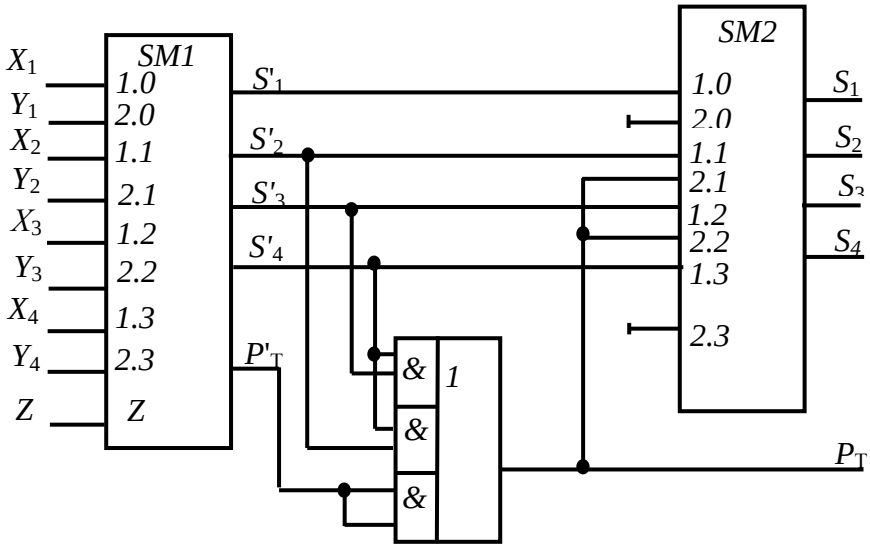


Рис. 1 Однорозрядний двійково-десятковий суматор

На рис. 2 наведено результати моделювання вказаного суматора.

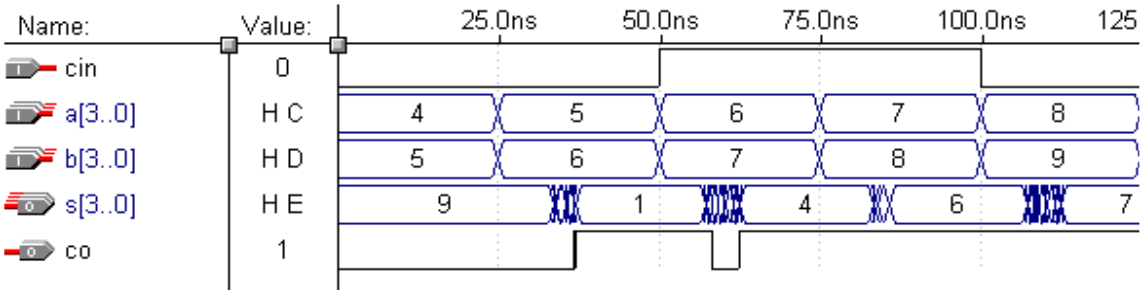


Рис. 2 Моделювання роботи однорозрядного двійково-десяткового суматора з використанням *D*-коду 8421

Однорозрядні суматори, які виконують додавання значень *i*-их розрядів A_i та B_i десяткових чисел з урахуванням переносу *Cin* з молодшого сусіднього розряду та виробляє на виходах функції результат S_i і перенесення *Cout* в старший розряд. Однорозрядний двійково-десятковий суматор з використанням *D*-коду 8421 реалізований за допомогою мови опису апаратних засобів *AHDL*, символ якого наведено на рис. 3.

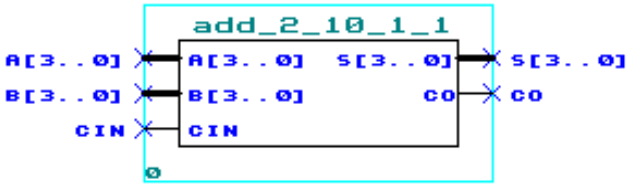


Рис. 3 Символ однорозрядного двійково-десяткового суматора з використанням *D*-коду 8421

На основі даного суматора на три входи та два виходи були побудовані суматори розрядністю в 4, 16, 32, 48, 64 тетради.

Введемо позначення $S'_T = S'_4 S'_3 S'_2 S'_1$ – двійкова тетрада після першого кроку додавання, за допомогою двійкового суматора SM1. Маємо три випадки:

1. Для значень $0 \le S'_T \le 9$ корекція не потрібна.
2. Для значень $10 \le S'_T \le 15$ потрібно відняти з одержаної суми число 10_{10} і здійснити перенесення в старшу сусідню декаду. Віднімання числа 10_{10} в доповняльному коді відповідає додаванню за допомогою двійкового суматора до попереднього результату числа шість, тобто плюс 0110_2. Ознакою такої корекції є одиничне значення функції корекції суми та перенесення:

$$F'_T = S'_4 \,\&\, S'_3 \# S'_4 \,\&\, S'_2;$$

3. Для значень $16 \le S'_T \le 19$ на виході суматора виникає перенесення P'_T з вагою 16_{10}. Однак у старшій декаді його значення сприймається як 10_{10}, тому потрібно додати до попереднього результату за допомогою суматора число шість, тобто 0110_2.
4. З урахуванням рівняння функції корекції результату та перенесення можна записати у вигляді:

$$P_T = P'_T \# F'_T = P'_T \# S'_4 \,\&\, S'_3 \# S'_4 \,\&\, S'_2.$$

Таким чином, у всіх випадках, коли $P'_T = 1$, до попередньої суми додається число 0110_2 і формується перенесення у старший розряд.

Таблиця 1. Десятковий еквівалент тетради

Десяткова сума	Сума після першого етапу $S'_5 S'_4 S'_3 S'_2 S'_1$	Правильна сума $PS_4\, S_3\, S_2\, S_1$	Корекція $C_5\, C_4\, C_3\, C_2\, C_1$	Примітка
0	00000	00000	00000	
1	00001	00001	00000	

2	00010	00010	00000	$0 \le S'_T \le 9$ Корекція не потрібна
3	00011	00011	00000	
4	00100	00100	00000	
5	00101	00101	00000	
6	00110	00110	00000	
7	00111	00111	00000	
8	01000	01000	00000	
9	01001	01001	00000	
10	01010	10000	00110	$10 \le S'_T \le 15$ Корекція потрібна: мінус 10 і перенесення в старшу тетраду
11	01011	10001	00110	
12	01100	10010	00110	
13	01101	10011	00110	
14	01110	10100	00110	
15	01111	10101	00110	
16	10000	10110	00110	$16 \le S'_T \le 19$ Корекція потрібна: плюс 6
17	10001	10111	00110	
18	10010	11000	00110	
19	10011	11001	00110	

Проілюструємо результат роботи 4-розрядного суматора на прикладі додавання Z = X + Y в коді 8421:

X = 1645_{10} = 0001 0110 0100 0101_2 ;

Y =2 275_{10} =0010 0010 0111 0101_2;

Z = 3920_{10} = 0011 1001 0010 0000_2.

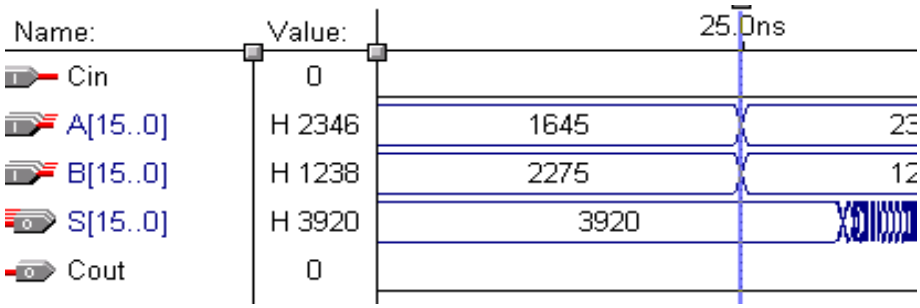


Рис. 4 Моделювання роботи 4-розрядного двійково-десяткового суматора з використанням *D*-коду 8421

Явна перевага суматора з використанням *D*-коду 8421 полягає у малих апаратних витратах на його побудову. Багаторозрядний суматор складається з такої кількості однорозрядних суматорів, яка дорівнює кількості розрядів тетрад, що додаються. Сигнал переносу

передається від розряду до розряду послідовно, утворюючи на виході значення старшого розряду суми.

Нижче наведено таблицю залежності затримок передачі даних від розрядності суматора (табл. 2), а також графік затримок (рис. 5).

Таблиця 2. Залежність затримки передачі даних від розрядності суматора.

Розрядність (тетради)	Суматор з використанням коду 8421
1	13,5
4	40,1
16	150,4
32	281,9
48	462,2
64	602,4

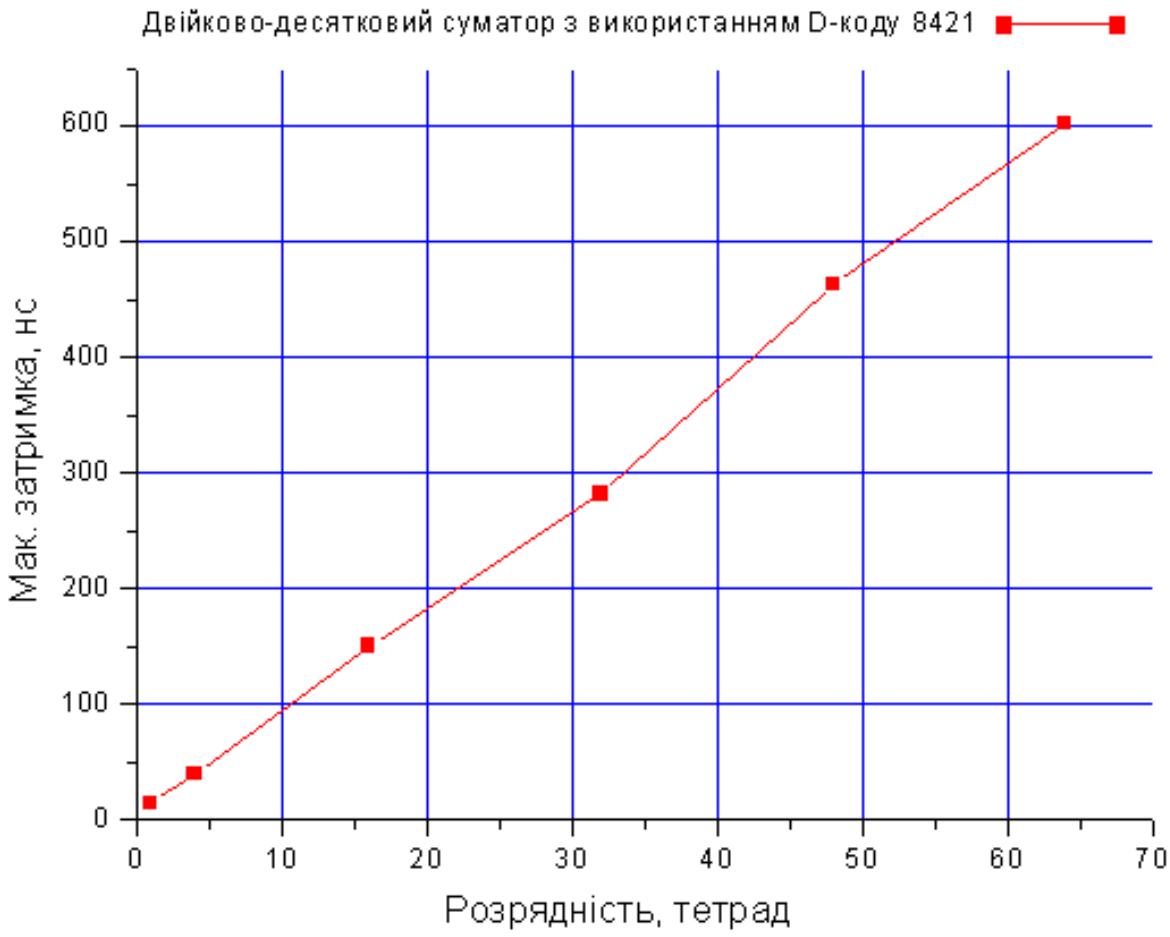


Рис. 5 Графік залежності затримки передачі даних від розрядності схеми.

Висновки

На основі аналізу схем багаторозрядних суматорів, в основу яких покладено максимальну затримку надходження даних на вихід, встановлено, що максимальна затримка роботи суматора збільшується майже лінійно.

Відмітимо, що було розроблено однорозрядний двійково-десятковий суматор з використанням D-коду 8421 мовою *AHDL*, на основі якого відбувалось подальше нарощення розрядності з допомогою *САПР MAX+plus II*.

Проаналізувавши схеми при виконанні арифметичних операцій у багаторозрядних суматорах, слід зазначити, що пристрої мають свої певні недоліки та переваги. Зокрема вони мають просту апаратну реалізацію, що прискорює швидкодію роботи суматора.

Підвищення швидкодії схем забезпечується за рахунок обробки масивів десяткової інформації, при чому вилучаються витрати часу на переведення чисел з десяткової системи числення в двійкову і навпаки.

Література

1. Основы языка AHDL / Ю.Ф. Опадчий // Язык описания аппаратуры AHDL. –2005. № 1. – С. 27- 35.
2. Палагин А.В. Реконфигурируемые структуры на ПЛИС / А.В. Палагин, В.Н. Опанасенко, В.Г. Сахарин // УсиМ. – 2000. – № 3. – С. 33–43.
3. Король І.Ю. Мова опису апаратури AHDL: Навчальний посібник для студентів напрямку 6.050102 – «Комп'ютерна інженерія»/ І.Ю. Король–Ужгород: УжНУ, 2008. –104 с.
4. Потемкин И.С. Функциональные узлы цифровой автоматики / И.С. Потемкин. – М.: Энергоатомиздат, 1988. – 320 с.
5. Глушков В.М. Синтез цифровых автоматов / В.М. Глушков. – М.: Физматгиз, 1996. – 467 с.
6. Биков М.М. Основи теорії цифрових автоматів / М.М. Биков. – Вінниця :ВНТУ, 2007. – 256 с.

РОЗРОБКА ЗАХИЩЕНОЇ ІНТЕРАКТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ “ВІРТУАЛЬНИЙ УНІВЕРСИТЕТ”

Кузьма Р.В.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Волошина, 54

Розвиток інформаційних технологій в наш час набирає дуже швидкі оберти. Звісно, це дуже вплинуло і на систему навчання як в Україні так і за її межами. На сьогоднішній день наступає новий етап ефективного застосування комп'ютерних навчальних систем. Вже відходить до другого плану паперова інформація, а їй на зміну приходять комп'ютери з локальними мережами, тому метою даної роботи є розробка моделі захищеної інтерактивної інформаційної мережі багаторольового доступу “Віртуальний університет”.

Велика увага в розробці данного порталу приділена аспекту комунікації студентів та викладачів між собою. А саме наступним чином: створення онлайн розкладу проведення пар та автоматичне сповіщення про його зміни, створення поштової системи, чату, онлайн оцінювання успішності та відвідування, розробка стрічки новин, онлайн обмінник файлів та університетська бібліотека, онлайн навчання та тестування знань, інтерактивний пошук та багато іншого. Також велика увага буде приділена саме захисту системи від різноманітного виду атак, або несанкціонованого доступу.

Мережа буде універсальною і доступною для реєстрації кожному університету. За структурою буде поділятися на студентську, викладацьку та адміністративну підсистеми з підсистемами.

Можливі ролі доступу

1. Адміністрація ректорату
2. Адміністрація деканату.
3. Викладач.
4. Студент(староста, заступник старости).
5. Батьки.
6. Підписники
7. Гості сайту

1. Адміністрація ректорату

Основні можливості адміністратора:

- Інформація про всіх студентів та персонал. А саме: телефон, адреса, особиста інформація, відбиток залікової книжки, набрані бали під час ЗНО, держ-заказ чи платник.
- Успішність, відвідування студентів або інша інформація у вигляді графіків(наприклад співставлення студентів на держ-заказі та платників, співставлення хлопців до дівчат). Причому як загальна, так і розподілена по факультетах
- Адміністрування загальнодоступної інформації про університет, що презентується на сайті – це інформація про факультети, кафедри, загальнодоступна бібліотека, інформація для абітурієнта, організація стрічки новин тощо
- Статистика та порівняльна характеристика по різних університетах та факультетах

2. Адміністрація деканату

- Адміністрування розкладу для студентів. Можливість розпечатки.
- Реєстрація студентів та персоналу
- Адміністрування дошки об'яв факультету або окремих спеціальностей. Зокрема, автоматизація і шаблонізація часто вживаних обяв і документів, таких як список дисциплін та їх годин, список заліків, диф. заліків та екзаменів, їхні дати і оприлюднення цієї інформації. Список студентів-боржників тощо.
- А також можливість швидкої загрузки цих документів і їхньої розпечатки на папері.
- Ведення статистичних даних, таких як успішність, відвідуваність студентів тощо.

3. Студент

- Розклад для групи
- Перегляд оцінок
- Персональні сповіщення(в тому числі на пошту) про зміну розкладу, виставлення нових оцінок тощо
- Перегляд відвідування для групи. Можливості старости
- Доступ до онлайн бібліотеки. Додавання своїх ресурсів
- Сторінка студента. Публічна інформація про себе

- Можливість роздруківки даних

4. Викладач

- Персональний розклад
- Річний план
- Ведення успішності і відвідування студентів
- Створення персональних тестів і можливість представлення їх студентам – можливість розпечатки. Процедура рішання тестів
- Статистика даних

Загальні можливості для ректорату, деканату, викладачів та студентів

- Створення новин і оголошень, навчального матеріалу
- Створення опитувань
- Ведення блогу
- Персональний та груповий чат
- Експорт / Імпорт даних

4. Батьки

- Перегляд інформації про успішність студентів
- Комунікація з персоналом факультету

5. Гості сайту(незареєстровані користувачі)

- Перегляд загальнодоступної інформації:
- Новини
- Інформація про факультет, персонал, процес навчання
- Оголошення
- Загальнодоступна бібліотека
- Інформація абітуріенту
- Наукова діяльність університету

6. Підписники

- Мають такі ж права як і гості, за виключенням того, що пройшли просту реєстрацію і отримують новини на пошту.
- Надання запиту на створення новини на дошці факультету

Онлайн навчання

- Створення курсів для студентів та абітурієнтів та можливість на них підписатися
- Створення відеолекцій, інтерактивних тестів, накопичення балів, доступ до навчальної мліератури

Розробка захисту на порталі

Основні методи і напрямки захисту:

- Розмежування доступу
- Капчі
- Вмираюча сесія
- Відправлення листа з підтвердженням реєстрації
- Валідація на складність паролю
- Хешування паролів
- Захист від SQL-ін'єкцій
- Захист від XSS атак

Наразі існуючі схожі проекти

Так, на даний час існують схожі розробки, які охоплюють вище сказані аспекти. Але, кожна з них тільки частково, тобто вони вузькоспеціалізовані. Є такі, що після їхнього встановлення потребують великого корегування і підстроювання під свої потреби, або ж просто незручні, чине мають української локалізації.

Дана розробка буде включати кращі аспекти, які зустрічаються в уже існуючих системах, а також запропоновані інші необхідні для навчання речі і об'єднані в одну інтерактивну систему. Отже порівняння з відповідними аналогами уможливлює окреслення переваг "Віртуального університету", а саме: універсальність використання, простота доступу, модульність, надійність.

Список деяких існуючих схожих систем:

- http://raspisaniye-vuzov.ru/ - розклад занять
- http://dls.kherson.ua/DLS/ - Херсонський Віртуальний Університет
- http://ubgd.lviv.ua/moodle/- навчальне середовище Львівського державного університету безпеки життєдіяльності
- http://vu.net.ua/ru/ - система дистанційного навчання
- https://moodle.org/ - ще одна широко відома система дистанційного навчання MOODLE

Порівняльна характеристика з існуючими системами

Розроблювана система торкається аспектів кожної існуючої системи. В ній зібрані основні їхні моменти. Виділяють її з поміж інших наступні пункти:

- Комплексність
- Легкий для розуміння інтерфейс
- Можливість автоматизації обробки інформації
- Універсальність
- Підтримка різних мов(зокрема: українська, російська, англійська)
- Підтримка смартфонів, планшетів
- Можливість створення нових університетів на основі файлу конфігурації з своєю базою даних

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ШИФРУВАННЯ ПОВІДОМЛЕНЬ У СОЦІАЛЬНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ КРИПТОГРАФІЧНИХ І СТЕГАНОГРАФІЧНИХ АЛГОРИТМІВ

Гарагонич Р. П., Спачинський В. В.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Волошина, 54

Соціальні мережі є одним із найзручніших засобів спілкування в Інтернеті, хоча захищеність особистої переписки стоїть під питанням, тому є необхідним розробити методи, які б вводили додатковий рівень безпеки.

У даній роботі ми розробили програму для обміну миттєвими повідомленнями, яка може функціонувати в facebook, ВКонтакте та в інших соціальних мережах. Основною ідеєю, яку ми перед собою ставили, є розробка такого алгоритму шифрування, який приховує у відкритому тексті зашифроване повідомлення.

Стеганографічні методи приховування текстової інформації у тексті мають погану стійкість до різних статистичних аналізів. Для підвищення стійкості використовують криптографічні алгоритми, які базуються на важко розв’язувальних математичних задачах. Тому для шифрування ми використовуємо алгоритм Гольдвассера-Блюма. Це семантично стійкий алгоритм, який базується на генераторі псевдовипадкових чисел BBS. Генератор ймовірнісним чином впливає на процес шифрування, тобто шифротекст сам по собі не містить ніякої корисної інформації.

При написанні програми ми використовували мову програмування – java. Це високопродуктивна об’єктно-орієнтована мова програмування, яка широко використовується при розробці програмного забезпечення для серверів.

Програма розробляється для мобільних пристроїв на базі операційної системи андроїд. Вона використовує відкрите API соціальної мережі ВКонтакте для обміну повідомленнями, фото- і відеофайлами.

Першим кроком ми реалізували обмін текстовими повідомленнями. Алгоритм роботи програми наступний.

Для кожного нового діалогу створюється набір ключів. Обмін ключами між користувачами може здійснюватися у як відкритому так і у прихованому вигляді.

Перед шифруванням таємне повідомлення перетворюється на бітову послідовність. Якщо використовувати стандартне кодування, яке використовується у java, то кожен символ буде займати 16 біт., а при розширеному наборі 32. Для оптимізації ми склали таблицю великих і малих літер українського алфавіту і пронумерували їх. Таким чином кожна літера представили 7 бітами. А розмір таблиці складає 128 символів, що цілком достатньо. Якщо знехтувати деякими буквами, то можна досягти розміру в 6 біт.

Функція шифрування і дешифрування має такий вигляд:

```
    public static CryptoText encryptionOrDecryption(boolean[] text,
BigInteger firstX, BigInteger n) {

       BigInteger random;
       boolean item;
       boolean[] cryptoText = text;

       for (int i = 0; i < cryptoText.length; i++) {
         random = firstX.multiply(firstX).remainder(n);
         firstX = random;
         item = random.remainder(TWO).toString().equals("1");

         cryptoText[i] ^= item;
       }
       return new CryptoText(cryptoText, firstX, n);
    }
```

Функція реалізує алгоритм шифрування Гольдвассера-Блюма. Для генерації випадкових чисел використовується алгоритм BBS. Для арифметичних операцій із великими числами ми використали вбудований у java клас BigInteger.

Цю функцію можна використати як для шифрування так і для дешифрування повідомлення. На вхід подається текст для шифрування (або для дешифрування), початковий член псевдовипадкової послідовності і відкритий ключі.

Для дешифрування потрібно визначити початковий член псевдовипадкової послідовності (ПВП) на основі закритого ключа і останнього члена ПВП.

```
private BigInteger getX0(int m, BigInteger randomKey) {
  Euclidean params = extendedEuclid(p, q);
  BigInteger a = params.x;
  BigInteger b = params.y;

  BigInteger l=p.add(ONE).divide(FOUR).pow(m).remainder(p.subtract(ONE));
  BigInteger k=q.add(ONE).divide(FOUR).pow(m).remainder(q.subtract(ONE));

  BigInteger u = randomKey.remainder(p).
                 pow(Integer.parseInt(l.toString())).remainder(p);
  BigInteger v = randomKey.remainder(q).
                 pow(Integer.parseInt(k.toString())).remainder(q);

  BigInteger mulA = a.multiply(p).multiply(v);
  BigInteger mulB = b.multiply(q).multiply(u);

  return mulA.add(mulB).remainder(n);
}
```

При стеганографічному приховуванні використовується метод символів однакового написання. Класичний алгоритм символів однакового написання використовує лише латинські букви для приховування, тому ефективність його роботи є досить низькою.

Наступним кроком є складання алфавіту. Кожній букві українського алфавіту ставиться у відповідність символ із таблиці UTF за візуальною подібністю. Тобто українську букву “а” легко можна замінити латинською буквою “a”, яка виглядає аналогічно. Потім відбувається співставлення відкритого повідомлення із зашифрованою бітовою послідовністю. Якщо біт $b_i = 1$ то літеру відкритого повідомлення, якій співставлений цей біт, замінюємо на символ із кодового алфавіту, якому відповідає ця літера. Якщо біт $b_i = 0$ то ми його пропускаємо.

Наступний фрагмент коду демонструє процес приховування таємного повідомлення. На вхід методу подаються два параметри: str – відкритий текст, encrypted – зашифроване повідомлення, яке представляється у вигляді масиву булевих значень.

```
public String encryption(String str, boolean[] encryptedText){

  String newStr = "";
  char[] arrayStr = str.toCharArray();

  int length = encryptedText.length;
  for (int i = 0, k = 0; i < arrayStr.length; i++) {
    char letter = arrayStr[i];
```

```
    char c = alf.get(letter)!= null ? alf.get(letter):0;

    if (c != 0 && k < length){
      newStr += encryptedText[k] ? c : letter;
      k++;
    }
    else{
      newStr += letter;
    }
  }
  return newStr;
}
```

Метод вертає рядок, який містить зашифроване повідомлення. Хоча візуальний вигляд повідомлення не змінюється, але при детальному обстеженні можна визначити підмінені літери.

Процес дешифрування проходить аналогічно, лише ключами в алфавіті виступатимуть не українські літери, а символи таблиці UTF.

Крім обміну текстовими повідомлення, можна використати стеганографічні алгоритми для приховування інформації в фото і відео файлах. У цих «контейнерах» можна зберігати значно більші обсяги інформації, ніж у текстовому повідомленні, а також вони забезпечується краще приховування інформації.

Отже, внесення додаткового рівня захисту в існуючі соціальні мережі є важливим завданням. Наша програма дозволяє шифрувати повідомлення і передавати їх в такому вигляді, який виключає підозри наявності криптограми у відкритому тексті.

Література:

1. Современная криптография: теория и практика. : Пер. с англ. – М. : Издательский дом “Вильямс”. 2005. – 768 с. : ил. – Парал. тит. англ.

РОЗРОБКА СИСТЕМИ ЕЛЕКТРОННОГО ОБЛІКУ УСПІШНОСТІ СТУДЕНТІВ

Дебич В.П., Бідюк Т.Ю.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Волошина, 54

Закономірним продовженням процесу інтеграції вітчизняної освіти у європейський освітній простір є впровадження кредитно-модульної системи у вищих навчальних закладах України. Підрахунок бально-рейтингової оцінки кожного студента містить велику кількість інформації, яку ретельно повинен обробити викладач. Цей факт особливо «напружує» викладачів, які звикли до «класичної» системи оцінювання. У таких умовах при великій завантаженості викладачів значний обсяг навчального часу треба витратити на оцінювання. Для економії часу викладача, запобігання технічних помилок та для полегшення умов доступу студентів до особистих даних з результатами їх оцінювання, нами було запропоновано розробку системи електронного обліку успішності студентів.

Ця система значно полегшує роботу не тільки викладачів, а й людей, які відповідають за створення розкладів екзаменаційних сесій та модульних контролів. Пропонуємо розглянути в чому полягає робота даної системи і які її можливості та переваги.

Почнемо з того, що при вступі кожен абітуріент повинен здати стандартний перелік основних документів, що засвідчують його особу та освітній статус. Ці дані під час зарахування на певний факультет заносяться до загальної електронної бази даних відповідного факультету.

Наступним кроком є те, що з вже створеної бази даних проходить генерація списків окремих курсів та груп студентів. Тобто, будь-який студент може відкрити головну сторінку і в запропонованому меню вибрати для себе відповідний курс та групу. З допомогою цієї функції користувач може переглянути список студентів відповідної групи і заочно познайомитись з людьми, які до неї входять. В випадку коли студент переводиться з однієї групи в іншу, за певними причинами, чи і зовсім відчисляється з факультету,

в базі даних можна внести зміни і система автоматично відредагує списки.

Ще однією дуже важливою функцією є створення загального розкладу екзаменаційних і залікових сесій, а також розклади модульних контролів. В даному меню студенти можуть як переглянути заплановані дати проведення модульних контролів, заліків, екзаменів з певних дисциплін, так і вносити певні пропозиції щодо зміни в розкладах. Система також обробляючи внесені пропозиції, не допускає збігів у розкладі. Тобто, якщо один і той же викладач веде різні дисципліни, то система не дозволяє проведення запланованих занять в один час, а корегує розклад.

Варто зазначити, що ще одним завданням для даної системи є автоматичне створення електронної відомості для кожної дисципліни і окремої академгрупи. Дана відомість автоматично генерується за допомогою списків груп, а також з створених на даний навчальний семестр розкладів модульних контролів, заліків і екзаменів з відповідних дисциплін. Всі бали отримані студентами під час модульних контролів та загальної оцінки заліку чи екзамену вносяться викладачем до вищезгаданої електронної відомості. Вносити зміни до даної відомості має можливість лише викладач, який веде дану дисципліну і ця процедура проходиться лише одноразово. Для підтвердження того, що інформація про оцінювання студентів є достовірною, тобто, що виставлені викладачем бали є реальними, електронні відомості з кожної дисципліни підписуються викладачем за допомогою електронно-цифрового підпису. Також системою передбачений пошук потрібної відомості за датою створення, назвою дисципліни або прізвищем викладача, який виставляв оцінки внесені до цієї відомості.

Заключним кроком в переміщенні оцінок є автоматичне перенесення всіх балів в електронну залікову книжку, індивідуальну для кожного студента. Залікова книга генерується окремо для кожного студента. Всі необхідні особисті дані до неї вносяться з бази даних. Оцінки з кожної дисципліни, яку здавав студент автоматично переносяться з залікових чи екзаменаційних відомостей. Доступ до особистої залікової книги матиме лише безпосередньо її власник, тобто доступ є обмеженим і всі занесені до неї дані не підлягають публічному перегляду. Оскільки відомості з оцінками захищені і не підлягають несанкціонованому редагуванню, а система працює в

автоматизованому режимі, то помилок в заліковій книзі бути не може.

Ми вважаємо, що досить цікавим і доцільним є впровадження в систему ще кількох пропозицій. Однією з них є введення різних графічних статистик. Тобто, на основі даних, які знаходяться в системі, можна створювати різноманітні графіки, діаграми різного роду статистики і порівняння. Наприклад, графіки успішності студентів різних груп за загальним балом підрахованим по всіх дисциплінах, або графік успішності між студентами однієї групи з окремої дисципліни і т. п.

Ще однією цікавою пропозицією є створення рейтингового списку «Топ 100». В цьому списку будуть відображатися імена студентів факультету з найвищим рейтингом на даний навчальний семестр. Система визначатиме середній бал кожного окремого студента факультету по успішності з усіх дисциплін і відповідно ті, чий бал буде найвищим і будуть відображатися у списку «Топ 100». На нашу думку дана невеличка функція стане стимулом для студентів до кращого вивчення дисциплін з метою одержання вищих балів.

На нашу думку, переваги електронної системи обліку успішності студентів полягають у наступному:

- всебічний поточний контроль та динамічне спостереження в будь-який момент навчання;
- забезпечення прозорої системи оцінювання студентів;
- економія часу викладача та запобігання помилок при виставленні балів;
- можливість швидкої обробки результатів та своєчасного інформування студентів та співробітників деканатів;
- можливість проведення статистичної обробки отриманих результатів сучасними методами.

КРИПТОГРАФІЧНА МОДЕЛЬ ЗАХИСТУ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ МЕТОДІВ ШИФРУВАННЯ ТА СТЕГАНОГРАФІЇ ДЛЯ ЗАБЕЗПЕЧЕННЯ ТАЄМНОСТІ ПЕРЕДАЧІ ПОВІДОМЛЕННЯ

Коцур М.М., Лишак С.Т.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Волошина, 54

Передача повідомлень завжди була важливою та важкою справою, особливо коли повідомлення містили таємну інформацію, тому у всі часи в різних куточках світу виникла одна і та ж сама проблема: «як зберегти таємність інформації, яка передається повідомленням?». Так, для вирішення цієї проблеми, почали використовувати криптографічні перетворення, які зручніше називати *криптографією*.

Взагалі, криптографія являє собою науку про методи та способи перетворення (шифрування) інформації з метою її захисту від несанкціонованих користувачів. Вона містить безліч методів, які використовуються не тільки для захисту інформації, але й для компактного представлення інформації, яку потрібно передати по сучасних каналах зв’язку. Дані методи можна розділити на чотири великі групи, кожна з яких має власну специфіку перетворення інформації, такі як:

1. Шифрування – процес проведення зворотних математичних, логічних, комбінаторних та інших перетворень вихідної інформації в результаті яких зашифрована інформація являє собою хаотичний набір символів.
2. Стеганографія – процес приховування не тільки змісту збереженої чи переданої інформації, але й сам факт зберігання чи передачі закритої інформації.
3. Кодування – процес заміни змістовних конструкцій вихідної інформації кодами.
4. Стиснення – процес зменшення об’єму інформації шляхом усунення надлишковості при кодуванні.

Але, задля ефективного криптографічного захисту інформації необхідно використовувати хоча б декілька методів криптографічного перетворення інформації, щоб забезпечити комплексність криптографічного захисту інформації від несанкціонованого ознайомлення чи модифікації, тобто потрібно створити модель криптографічного захисту, яка б в собі комбінувала різноманітні методи перетворень з вище перерахованих груп криптографічних перетворень інформації.

Так, проаналізувавши спеціалізовану літературу щодо криптографічного захисту інформації, ми вирішили розробити власну модель криптографічного захисту інформації, що буде передаватися повідомленням. Власне кажучи, наша модель захисту полягає в тому, щоб інформацію, яку потрібно передати по сучасних каналах зв'язку, необхідно криптографічно перетворити за певним алгоритмом в шифрограму та приховати факт її передачі замаскувавши повідомлення в графічному файлі, тобто зашифроване повідомлення помістити всередину зображення, яке буде відігравати роль контейнера при передачі повідомлення між адресатами в обхід небажаних персон.

Звідси випливає мета нашої роботи, яку можна сформулювати так: «Розробити таку модель криптографічного захисту інформації, яка б забезпечувала таємність передачі повідомлення між адресатами замаскувавши його під виглядом зображення, яке б не привертало до себе уваги, та реалізувати програмний засіб, який здійснював би дані перетворення».

Поставивши перед собою завдання по розробці програмного засобу, який буде здійснювати перетворення відкритого тексту в шифрограму та подальшого поміщення в зображення, необхідно провести ряд експериментів із ін'єкції зашифрованого повідомлення в графічні файли, задля знаходження оптимального місця розташування шифротексту. Суть експериментів полягає в тому, щоб графічні файли відкрити через текстовий редактор типу блокнот і вручну поміщати короткі набори символів в різні частини зображення та в різних форматах графічних файлів. Так, в результаті узагальнення даних щодо проведених експериментів, можна сказати, що найкращим місцем розташування зашифрованого повідомлення в зображенні є останній рядок самого зображення, так як при цьому графічні файли не пошкоджуються та можуть переглядатися за

допомогою стандартних програм перегляду зображень без будь-яких проблем.

Спираючись на дані проведених досліджень можна приступити до практичної реалізації нашої модель, тобто розробка програмного засобу який здатний виконати поставленні завдання для даної моделі.

В ході розробки програмного засобу, ми вирішили реалізувати дві окремі програми, які будуть виконувати необхідні криптографічні перетворення інформації згідно нашої моделі. Так одна з цих програм буде виконувати алгоритм шифрування для відкритого тексту, а сам результат даної операції буде поміщати в останній рядок зображення. Інша ж програма буде виконувати зворотні криптографічні перетворення, тобто вибиратиме останній рядок із зображення та виконувати подальше дешифрування шифрограми.

Тепер більш детально зупинимось на програмі шифрування, ми її назвали ***crypt.rb***. Цей програмний засіб був написаний на мові програмування **Ruby**, яка забезпечує гнучкість при розробці компактних програмних продуктів, та використовує алгоритм шифрування простої однолітерної заміни для латинської абетки. Для того, щоб дана програма здійснювало криптографічне перетворення необхідно відкрите повідомлення записати латинськими літерами в наперед підготовленому текстовому файлі з якого під час виконання програми буде зчитуватися повідомлення та вказати ім’я графічного файлу з його розширенням. В результаті в зображення буде записана шифрограма, а саме зображення не пошкодиться. Код програми наведено в лістингу 1.

Лістинг 1. Код програми ***crypt.rb***

```
Start=File.open("crypt.txt“, “r+”)          //зчитування повідомлення з
open_text=start.readline                    //текстового файлу
start.close



text=open_text.split(//)                    //по-символьне розбиття
text.each do |sym|                          //рядку
```

```
        num = alfa.index(sym)                    //шифрування повідомлення
        str = beta[num]
        close_text += str
end

file=File.open("#{image}", "a+")                 // запис повідомлення в
file.print “\n”+close_text                       //кінець зображення
file.close
```

Програма **decrypt.rb** виконує зворотні перетворення і має аналогічний код. Ці програмні засоби були реалізовані задля демонстрації дієздатності нашої розробленої моделі, так як в результаті криптографічного перетворення інформації графічні файли не були пошкодженні і могли відтворюватись без проблем в різних програмах для перегляду зображень. В програмах був реалізований алгоритм шифрування простої однолітерної заміни, хоча можна замінити даний алгоритм на будь-який інший та використовувати будь-яку абетку чи іншу сукупність символів.

Таким чином можна зробити висновок, що наша модель забезпечує таємність передачі повідомлення не тільки із-за використання алгоритмів криптографічних перетворень, але й маскування шифрограми в зображенні, яке не буде привертати до себе уваги.

Література

1. Хорошко В.А., Чекатов А.А. Методы и средства криптографической защиты информации / Под ред. Ю.С. Ковтанюка. - К.: Изд-во Юниор, 2003. - 504с.
2. Мухачев В.А., Хорошков В.А. Методы практической криптографии. - К.: ООО «Полиграф-Консалтинг», 2005.-215с.

ПРОГРАМНІ ЗАСОБИ УЩІЛЬНЕННЯ ВІДЕОДАНИХ

Середній В.Я.

ДВНЗ „Ужгородський національний університет”
88000, Ужгород, вул. Університетська, 14

У сучасних умовах мультимедійна інформація створюється, накопичується, зберігається на цифрових носіях та передається каналами зв’язку практично всіма користувачами комп’ютерних систем. За рахунок програмних засобів можна істотно підвищити швидкість обміну даними через мережі та зменшити обсяги використання дискового простору, виконуючи ущільнення відеоданих.

Термін відео (від лат. *Video* − дослівно «бачу») охоплює широкий спектр технологій запису, обробки, передачі, зберігання та відтворення візуального і аудіовізуального матеріалу. Відеодані – це по суті тривимірний масив кольорових пікселів. Два виміри означають вертикальний і горизонтальний розміри кадру, а третій вимір – момент часу. Кадр – це масив усіх пікселів, які розміщені перед камерою в певний момент часу, або просто зображення. Ущільнення відео – це зменшення кількості даних, які використовуються для подання відеопотоку.

Ущільнення базується на наявності:

1. Просторової надлишковості інформації, присутньої в кожному кадрі.
2. Часової надлишковості − переважна кількість кадрів є подібними до попереднього і / або наступного кадрів.

Ущільнення відео було б неможливе, якби кожен кадр був унікальним, а кольори пікселів – повністю випадковими, але це не так. Тому можна ущільнювати, по-перше, саме зображення – наприклад, фотографія блакитного неба без сонця фактично зводиться до опису граничних точок і градієнта заповнення. По-друге, можна ущільнювати схожі сусідні кадри. Зрештою, алгоритми ущільнення зображень і відеоданих подібні, якщо розглядати відео як тривимірне зображення з часом як третьою координатою.

Більшість відомих алгоритмів, які дозволяють ущільнити відео в десятки разів (*M-JPEG, MPEG-1, -2, H.264*), призводять до

незначних втрат якості відео, які не помітні для ока людини при перегляді. Крім цього, існують алгоритми, які ущільнюють без втрат якості (*CorePNG, GIF89a*), але ступінь ущільнення є дуже малий. В залежності від ступеня ущільнення змінюється і якість відео.

Сучасні алгоритми ущільнення використовують дискретне косинусне перетворення (*DCT – ДКП*) або його модифікації для усунення просторової надлишковості. Інші методи, такі як фрактальне ущільнення та дискретне вейвлет-перетворення, також були об'єктами досліджень, але зараз використовуються тільки для статичних зображень. Використання більшості методів ущільнення (таких, як дискретне косинусне перетворення та вейвлет-перетворення) спричиняє також використання процесу квантування. Квантування може бути як скалярним, так і векторним, тим не менше, більшість схем ущільнення на практиці використовують скалярне квантування завдяки його простоті.

ДКП є широко використовуваним при ущільненні зображень перетворенням. Стандарт ущільнення статичної графіки *JPEG*, використовуваний у відеоконференціях стандарту *H.263*, цифрові відеостандарти *MPEG* (*MPEG-1*, *MPEG-2* і *MPEG-4*) − всі вони використовують ДКП. У цих стандартах використовується, зокрема, двовимірне ДКП, використовуване послідовно до блоків зображення розміром 8×8 пікселів. ДКП обчислює 64 (8×8 = 64) коефіцієнти, які потім квантуються, забезпечуючи тим самим реальне ущільнення. У більшості зображень більшість ДКП коефіцієнтів через своє мале значення після квантування обнулюються. Ця властивість ДКП і лежить в основі багатьох алгоритмів ущільнення, які використовують ДКП.

Для ущільнення в багатьох алгоритмах застосовується векторне квантування, яке полягає в розбитті зображення на блоки (розміром 4×4 пікселі в колірній схемі *YUV* для компресорів *Indeo* і *Cinepak*). Як правило, деякі блоки виявляються схожими один на одного. В цьому випадку компресор визначає клас схожих блоків і замінює їх одним загальним блоком. Крім того, генерується двійкова таблиця (карта) таких загальних блоків з найкоротших кодових слів. *VQ*-декодер потім, використовуючи таблицю, збирає зображення поблочно із загальних блоків. Зрозуміло, що даний спосіб кодування з втратами якості, оскільки подібність блоків дуже відносна. Тут допускається апроксимація реальних блоків зображення до

загального, їх об'єднуючого. Процес кодування тривалий і трудомісткий, оскільки кодеру необхідно виявляти відповідність кожного блоку зображення до якого-небудь загального блоку. Проте завдання декодування в цьому випадку зводиться до завдання побудови зображення за заданою картою із загальних блоків і не займає багато апаратних і часових ресурсів. Таблицю або карту також називають ще і кодовою книгою, а двійкові коди, що входять в неї, − кодовими словами, відповідно. Найбільше ущільнення з використанням алгоритму *VQ* досягається шляхом зменшення числа класів загальних блоків, тобто припущенням про подібність відносно більшого числа блоків зображення, і як наслідок, зменшенням кодової книги. У міру зменшення розмірів кодової книги якість відтвореного відео погіршується. В результаті на зображенні з'являється штучна «блочність».

JPEG − практично є стандартом де-факто для повнокольорових зображень. Оперує алгоритм областями 8x8, на яких яскравість і колір міняються порівняно плавно. Внаслідок цього, при розкладанні матриці такої області в подвійний ряд по косинусах значимими виявляються лише перші коефіцієнти.

Алгоритм :

1. Переводимо зображення з колірного простору *RGB* в колірний простір *YCrCb*.
2. Розбиваємо вихідне зображення на матриці 8x8. Формуємо з кожної три робочі матриці ДКП − по 8 біт окремо для кожної компоненти кольору.
3. Застосовуємо ДКП до кожної робочої матриці.
4. Проводимо квантування. В принципі, це просто ділення робочої матриці на матрицю квантування поелементно.
5. Переводимо матрицю 8x8 в 64-елементний вектор за допомогою “зигзаг”-сканування (тобто беремо елементи з індексами (0,0) (0,1) (1,0) (2,0)...).
6. Згортаємо вектор за допомогою алгоритму групового кодування.
7. Згортаємо пари, що вийшли, кодуванням Хаффмана з фіксованою таблицею.

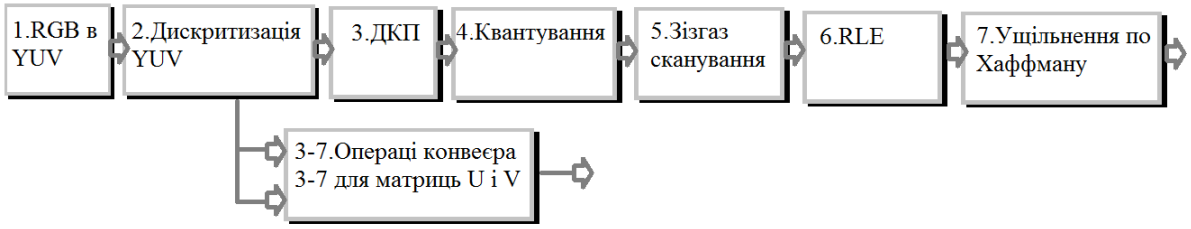


Рис.1 Конвеєр операцій, використовуваний в алгоритмі *JPEG*.

MPEG − це абревіатура від *Moving Picture Experts Group*. *MPEG-1* дуже популярний формат у всьому світі з основою, взятою від кодека *JPEG*. Ущільнення в ньому проводиться серіями по три кадри. Це один з найстаріших кодеків, так що, практично на будь-яких, навіть «найслабкіших» машинах можна проглянути відео із стереозвуком в цьому форматі. Проте і якість зображення невисока, вона порівнюється із звичайним аналоговим форматом *VHS*. Зображення має роздільну здатність 352x288 точок, а і якість його залишає бажати кращого. І хоча *MPEG-1* не вимогливий до ресурсів, його доля вирішена наперед: з розвитком обсягу і швидкості передачі даних в комп'ютерах і в Інтернеті формат поступово забуватиметься.

У форматі *MPEG-1* всі кадри відеоролика підрозділяються на три типи: I-, *P*- і *B*-кадри (рис. 2). До першого типа (*I*-кадри, *Intra Frames*) відносяться опорні кадри. Їх зображення зберігаються в повному об'ємі у форматі *JPEG*. Для *P*-кадрів (*Predicted Frames*) записуються лише відмінності від попереднього I-кадру, що вимагає набагато менше дискового простору. Для B-кадрів (*Bi-Directionally Interpolated Frames*) зберігаються відмінності від попереднього і наступного *I*- або *P*-кадру.

MPEG-2 є подальшим розширенням *MPEG-1*. У ньому збільшений розмір кадру, він складає 1920 x 1080 точок, додана підтримка шестиканального звуку. Проте для відтворення відео в цьому форматі потрібна вища обчислювальна потужність комп'ютера. *MPEG-3* призначався для використання в системах телебачення високої чіткості (*high-defenition television, HDTV*) із швидкістю потоку даних 20-40 Мбіт/с, але пізніше став частиною стандарту *MPEG-2* і окремо тепер не згадується. До речі, формат *MP3*, який інколи плутають з *MPEG-3*, призначений лише для ущільнення аудіоінформації і повна назва *MP3* звучить як *MPEG Audio Layer III*.

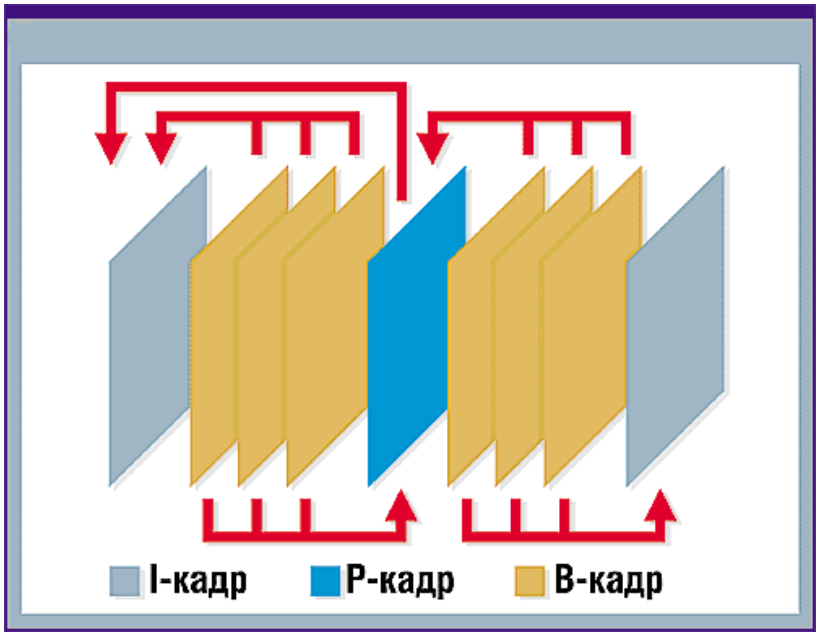


Рис. 2 *I*-, *P*- і *B*-кадри

Формати *MPEG-1* і *MPEG-2* не забезпечували реальної можливості трансляції відео по мережі Internet і створення інтерактивного телебачення на їх основі − дуже вже великим був розмір файлів. Для радикального зменшення, а також реалізації інших функцій, необхідних для передачі потокового відео, була почата робота над специфікаціями нового формату − *MPEG-4*. По суті, він орієнтований не стільки на ущільнення відео, скільки на створення так званого "мультимедійного контенту" − злиття інтерактивного телебачення, 3D-графики, тексту і так далі.

MPEG-4 поєднує відмінний звук і максимальне ущільнення відеосигналу (до 30-40% кращим чим у попередників). Різниця полягає в тому, що кодується послідовність більш ніж з трьох кадрів (звичайно до 250 кадрів). Тим самим досягається більше ущільнення і можливість перегляду в режимі реального часу якісного потокового відео в Інтернеті.

У роботі реалізовано ущільнення відеоданих на основі стандарту MPEG-4. Тестування розробки дозволяє зробити висновки, що реалізовані алгоритми дають можливість перекодувати відео у форматах MPEG-1, MPEG-2 та інших у формат MPEG-4 з якістю, яка не поступається комерційним розробкам.

Література

1. Юдін О.К. Структурно-логічна модель кодера стиску інформаційного потоку даних / Юдін О.К., Чеботаренко Ю.Б., Курінь К.О. // Вісник Інженерної академії України. – К., 2010. – 4 вид. – с. 151-157.

2. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. − М.: ДИАЛОГ-МИФИ, 2003. − 384 с.

3. Le Gall D. J. The MPEG Video Compression Algorithm // Signal Processing: Image Communication. 1992. Vol. 4, № 2. P. 129-140.