

М.О. Гаврилюк, О.В. Івахів\*, В.Я. Пуйда, П.М. Сасник\*  
Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин,  
\*кафедра приладів точної механіки

## ПІДСИСТЕМА ДЛЯ МОДЕЛЮВАННЯ АЛГОРИТМІВ ОБРОБКИ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ СИМУЛЯТОРА DSP-56300

© Гаврилюк М.О., Івахів О.В., Пуйда В.Я., Сасник П.М., 2004

Системи технічного зору реального часу часто будуються на базі ЦПОС. Розглянута підсистема дає змогу моделювати та відлагоджувати задані алгоритми обробки зображень, використовуючи симулятор DSP-56300, ще до створення реальних апаратних засобів, що дозволяє пришвидшити розробку та зменшити можливі помилки.

The Real Time Computer Vision Systems it is are often built on the base of DSP. The considered subsystem allows to design and debug the set algorithms of the image processing, using simulator DSP-56300, yet to creation of the real hardware equipment, that allows to accelerate development process and decrease possible errors.

**Вступ.** 80 % інформації про навколишній світ людина отримує через візуальне сприйняття, а тому під час створення інтелектуальних цифрових пристроїв їх дедалі частіше обладнують пристроями обробки зображень [1]. Такі пристрої широко використовуються в промисловості, зокрема лише в секторі систем технічного зору за даними асоціації постачальників та інтеграторів в області машинного зору Automated Imaging Association (AIA) світовий ринок впровадження СТЗ становить на даний момент \$8 млрд., а в найближчі два–три роки потроїться, в побутових приладах також все ширше використовуються пристрої опрацювання зображень: цифрові фото, відеокамери, мобільні телефони та ін.

При побудові сучасних систем опрацювання зображень розробка алгоритмів найчастіше ведеться за допомогою ПК, де відбувається їх симуляція та відлагодження. Широко поширене використання для таких цілей пакета Matlab. Проте для виявлення проблем із швидкістю використання ресурсів, апаратних несумісностей цього пакета недостатньо, потрібна система відлагодження, яка б працювала з реальними апаратними засобами, крім того, система повинна мати інтерфейс зв'язку з ПК, щоб можна було легко завантажувати модифіковані програми. Таким вимогам відповідає симулятор DSP56300 [4].

**1. Опис системи.** Ця система дає змогу змоделювати певний алгоритм обробки зображень, використовуючи симулятор DSP-56300.

До програмних засобів увійшли: симулятор Suite56™ DSP Simulator із стартового комплексу розробника, асемблер DSP56303[5], лінкер DSP56303[6], сі-компілятор для DSP56303, програма перетворення зображення з формату npt до формату зрозумілого симулятору lod, програми реалізації алгоритму обробки, та програми зворотнього перетворення форматів файлів з .lod у .npt. Вхідне зображення у форматі .npt формується чорно-білою відеокамерою. Зображення подається у вигляді масиву розміром 256\*256 кожен елемент якого є байтом ( напівтонове зображення).

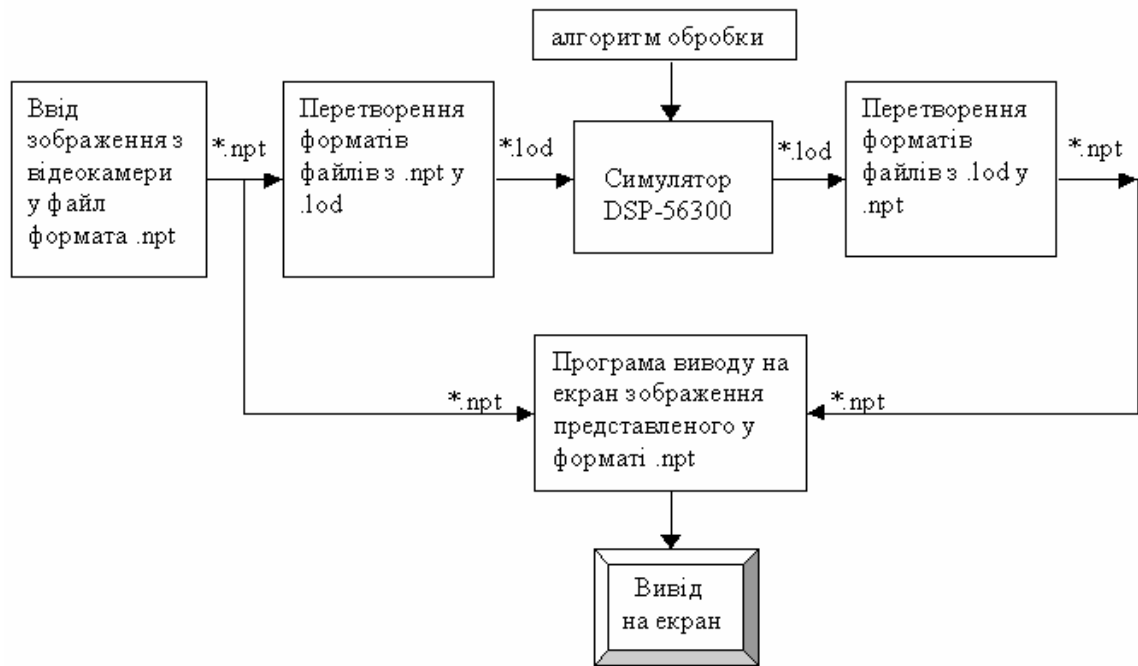


Рис. 1. Процес моделювання алгоритмів

**1.1. Процес моделювання.** У систему зображення вводиться чорно-білою відеокамерою і зберігається у файлі формату `npt`. Безпосередньо завантажити отримане зображення з відеокамери у форматі `.npt` в симулятор **DSP-56300** неможливо, оскільки дані у форматі `.npt` подані в упакованому вигляді. Дані, що безпосередньо можна завантажити у симулятор, повинні бути у форматі `.lod`. Дані у форматі `.lod` подані так:

```

_DATA X 000030
979A9A 9D9C9D 9C9E9E A1A2A4 A4A6A7 A9A8A9 ...
_END

```

Файл типу `lod` складається з трьох секцій: секції заголовку, яка складається з таких записів: запис `_DATA` вказує, що дані призначені до завантаження в пам'ять даних на кристалі процесора, запис `X`, в яку пам'ять (X чи Y) завантажувати дані та адреса з якої почати розміщення даних `000030`, другу секцію становлять безпосередньо самі дані, які записуються групами по 3 байти (6 символів). У першому рядку вказується пам'ять та її адреса з якої буде починатися масив. В другому рядку містяться безпосередні дані масиву. Кожні три байти відокремлені пробілом, оскільки комірки пам'яті в симуляторі **DSP-56300** 24-х розрядні. В кожній комірці пам'яті симулятора буде міститися інформація про три піксели. В останньому рядку міститься директива, що інформує симулятор про кінець файла, секція кінцівки із значенням `_END`.

Програма `npt_lod.exe` за допомогою якої здійснюється перетворення формату файла `.npt` у `.lod`. Ця програма працює за таким алгоритмом: з кожного байта вхідного файла `test.npt` створюється два байти вихідного файла `test.lod` утворенням з молодшої та старшої частин вхідного байта їх відповідних ASCII значень. Це здійснюється виділенням чотирьох бітів та додаванням до них необхідної суми для утворення відповідного ASCII значення (якщо число знаходиться в межах від 0 до 9 то додається 48, якщо від a до f то додається 55). А також після кожних шести записаних у вихідний файл байтів записується ASCII код пробілу у вихідний файл (це робиться для створення відокремлених слів для симулятора **DSP-56300**). Завантажене у пам'ять симулятора зображення обробляється за заданим алгоритмом і результат зберігається у файлі `test_out.lod`, за допомогою програми `lod_npt.exe` здійснюється зворотне перетворення з формату `.lod` до формату `.npt`, а потім спеціальною програмою здійснюється вивід на екран результуючого зображення.

**1.2. Алгоритм опрацювання зображення.** Для зменшення кількості даних, що опрацюються, вхідне напівтонове зображення перетворюється до бінарного формату [3]. Бінарні зображення кодують мінімальну кількість інформації в кожному пікселі [2]. В бінарному представленні зображення кожний відлік  $a_i$  набуває значення 0 або 1. Всі відтинки сірого умовно поділяються на два кольори: чорний, якому відповідають одиничні відліки, та білий, якому відповідають нульові відліки. Квантування відеосигналу зводиться до порівняння його амплітуди в кожній точці рецепторного поля з деякою пороговою величиною  $W$ . При використанні матричного кодування бінарні відліки щільно запаковуються у слова машинної пам'яті так, що для представлення кожного відліку відводиться тільки один біт інформації за один піксель і в кожному байті зберігається 8 пікселів зображення. Відповідно піксели згруповані по рядах зображення. Перший байт містить перші 8 пікселів в першому ряді зображення. Другий байт містить наступні 8 пікселів в першому ряді зображення і т. д. Практично, щоб почати наступний ряд, якщо кількість пікселів в ряді зображення не 8-кратне, біти, що залишилися в останньому байті, не використовуються. Кожен новий ряд зображення починається новим байтом. Ряди зображення вирівнюються по границі байта. Деякі системи можуть вимагати, щоб це вирівнювання було на границі великого групування, наприклад, з 32 або 16 бітами, що може бути зручним для виконання обробки відповідними засобами, в нашій системі це вирівнювання здійснюється на границю 24 біти.

Дослідження алгоритму перетворення напівтонових зображень в бінарні проведено за базовою формулою числового визначення лапласіана

$$G_0 = \begin{cases} 1, & \text{if } (f_0(x, y) > S) \text{ or } (f_0(x, y) = S \neq 0), \\ 0, & \text{if } (f_0(x, y) < S) \text{ or } (f_0(x, y) = S = 0), \end{cases}$$

де  $G_0(x, y)$  – результат перетворення (бінарне значення точки  $f_0(x, y)$ ),

$$S(x, y) = \frac{K}{m * n - 1} * \left\{ \left[ \sum_{x=1}^m \sum_{y=1}^n f(x, y) \right] - [f_0(x, y)] \right\},$$

де  $S(x, y)$  – середнє значення яскравості по околу (далі  $S$ ) (рис. 2);  $f(x, y)$  – значення яскравості точки з координатами  $(x, y)$ ;  $m, n$  – розмір околу точки;  $f_0(x, y)$  – яскравість центральної точки;  $K$  – ваговий коефіцієнт обчисленого середнього значення.

У нашому випадку розмір околу вибрано 3x3, а тому  $S$  обчислюється за формулою:

$$S = (f(x-1, y-1) + f(x, y-1) + f(x+1, y-1) + f(x-1, y) + f(x+1, y) + f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)) / 8,$$

де  $G(x, y)$  – бінарне значення піксела  $(x, y)$ ,  $f_0(x, y)$  – напівтонове значення піксела  $(x, y)$ .

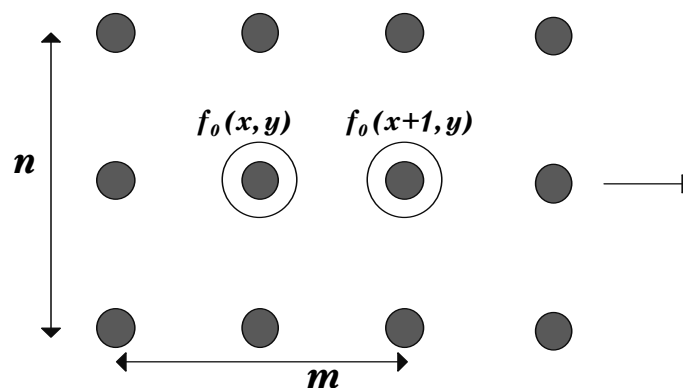


Рис. 2. Визначення бінарного значення точки

**Висновок.** На базі цієї системи можна ефективно відпрацьовувати алгоритми опрацювання зображень, відлагоджувати їх, а також проводити різнобічний аналіз створених алгоритмів (швидкодія,

витрати ресурсів) на засобах, наближених до реальних. За допомогою створеного додаткового програмного забезпечення отримано можливість проводити опрацювання реальних алгоритмів.

1. Э. Опенгайм. *Применение цифровой обработки сигналов*. – М.: Мир, 1980. – С. 192–267.
2. Фурман Я.А., Юрьев А.Н., Яшин В.В. *Цифровые методы обработки и распознавания бинарных изображений*. – Красноярск: Изд-во Краснояр. ун-та, 1992. – С. 12–19.
3. William K. Pratt. *Digital Image Processing: PIKS Inside*, John Wiley & Sons, 2001, Third Edition. – С. 150–159.
4. Motorola DSP Simulator reference manual, <http://www.motorola-dsp.com>.
5. Motorola DSP assembler reference manual, <http://www.motorola-dsp.com>.
6. Motorola DSP linker/librarian reference manual, <http://www.motorola-dsp.com>.
7. Motorola DSP56303 User's Manual, <http://www.motorola-dsp.com>.

УДК 681.3

В.С. Глухов

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## СИСТЕМА КОМАНД КРИПТОГРАФІЧНОГО ПРОЦЕСОРА

© Глухов В.С., 2004

**Описаний підхід до синтезу системи команд спеціалізованого процесора для створення і аналізу цифрового підпису.**

**The approach to synthesis of digital signature specialized processor instruction set is described.**

**Вступ.** Наш світ стає все більше електронним в тому сенсі, що багато сфер людської діяльності вже значною мірою пов'язані з цифровим світом. Технології змінюють сталу термінологію – вже звичними стали словосполучення: електронний документообіг, електронна комерція, електронний уряд. А з 1 січня 2004 року в Україні офіційно дозволено користуватися електронним цифровим підписом замість звичайного [1].

Сьогодні в Україні діють два стандарти на цифровий підпис: міждержавний стандарт ГОСТ 34.310–95 [2] та національний стандарт України ДСТУ 4145–2002 [3]. Поряд з великою кількістю літератури про основи і принципи захисту інформації [4–7, 16] останнім часом у фахових виданнях розглядаються конкретні питання впровадження систем захисту інформації, в тому числі і систем цифрового підпису, в нетрадиційних галузях [8], проектування структури цих засобів [9–11], питання стійкості закладених в них алгоритмів [12]. Разом з ними актуальними є також і розглянуті у цій статті питання організації роботи криптографічних процесорів. До цих питань зокрема належить розподіл функцій між центральним і спеціалізованим процесором та вибір системи команд останнього.

**Окреслення проблеми.** Нині загальноновизнаними є декілька способів побудови криптографічних систем. Одним з них є використання стандартних універсальних програмованих процесорів і процесорів обробки сигналів. Цей підхід характеризується невисокою швидкістю.

Перспективнішим є використання сукупності програмованого (центрального) процесора і спеціалізованого криптографічного процесора (або кількох спеціалізованих процесорів), при цьому спеціалізовані процесори можуть бути виконані у вигляді ядер НВІС [4, 9, 11]. При цьому актуальною стає задача організації ефективної взаємодії центрального і спеціалізованого процесорів, а також задача вибору системи команд останнього.

Тут описаний підхід до реалізації взаємодії програмованого (центрального) процесора з ядром спеціалізованого криптографічного процесора, який виконує дії, необхідні для отримання цифрового підпису.