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

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 загальновизнаними є декілька способів побудови криптографічних систем. Одним з них є використання стандартних універсальних програмованих процесорів і процесорів обробки сигналів. Цей підхід характеризується невисокою швидкодією.

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.

Огляд методів взаємодії програмованого (центрального) процесора і спеціалізованого процесора. Відомі декілька методів взаємодії центрального процесора і спеціалізованого:

1) спеціалізований процесор (сопроцесор) аналізує потік команд, які в ході виконання своєї програми зчитує з пам'яті команд центральний процесор, сопроцесор знаходить серед них ті, які входять до його власної системи команд, і виконує їх. За таким методом взаємодіють центральний процесор і математичний сопроцесор у мікропроцесорах ф. Intel;

2) центральний процесор пересилає до регістрів спеціалізованого процесора (сопроцесора) дані для обробки та інструкцію, яка вказує, що треба робити з цими даними. За таким методом взаємодіють центральний процесор і математичний сопроцесор у мікропроцесорах ф. Motorola;

3) центральний і спеціалізований процесори мають незалежні лічильники команд, системи команд, можуть виконувати свої програми паралельно. Центральний процесор може запускати спеціалізований процесор на виконання тієї чи іншої програми і потім приймати результати виконання цих програм. Спеціалізований процесор у цьому випадку можна назвати спецобчислювачем. Аналогом такого методу взаємодії є взаємодія центрального процесора і каналу вводу-виводу, тому спеціалізований процесор можна також назвати "математичним каналом" [13, 14].

Перший метод вимагає глибокого знання особливостей архітектури центрального процесора, його системи команд, використання її кодів для запуску сопроцесора. Як правило, такі сопроцесори можуть використовуватися тільки з одним типом центральних процесорів, що обмежує їхнє використання у вигляді ядер.

Другий і третій методи вільні від цих недоліків.

При глибшому розгляді можна побачити, що спецобчислювач (метод 3) містить (в тому чи іншому вигляді) центральний процесор нижнього рівня і сопроцесор (метод 2). Така багаторівнева структура і аналогія з каналом вводу-виводу дає змогу застосувати при проектуванні спецобчислювача еталонну модель взаємодії відкритих систем для розподілу функцій між елементами системи. При такому підході можна вважати, що сопроцесор і спецобчислювач виконують функції двох нижніх рівнів моделі: сопроцесор виконує функції "фізичного" рівня (виконує "пересилання" даних через свій арифметико-логічний пристрій з тими чи іншими властивостями), а спецобчислювач – функції канального рівня. На центральний процесор нижнього рівня покладається задача взаємодії з центральним процесором верхнього рівня і управління роботою сопроцесора (рис. 1).

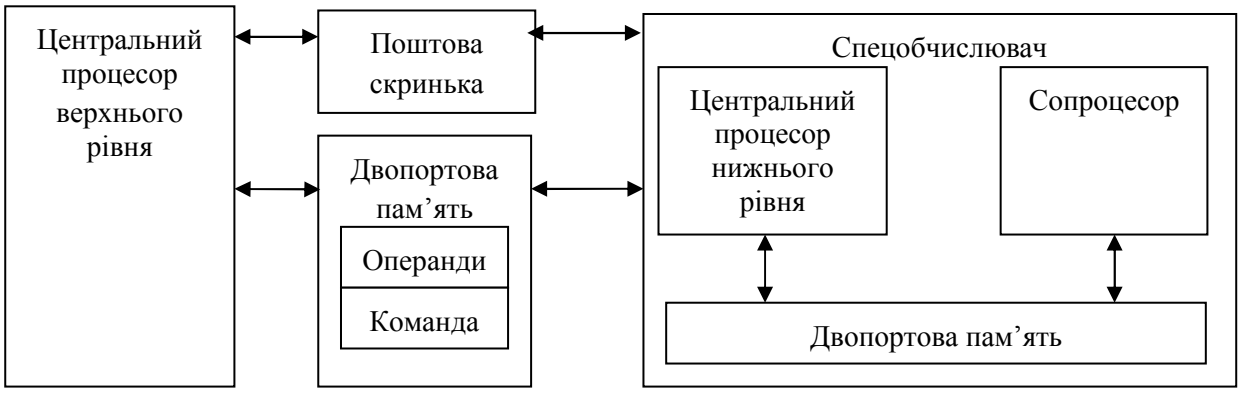

Рис. 1. Криптографічна система

Детальну структуру криптографічного спецобчислювача показано на рис. 2.

За необхідності (і не тільки для криптографічних систем, а для спеціалізованих систем взагалі) кількість рівнів може бути збільшена "вгору" (згідно з еталонною моделлю – до семи), при цьому вони будуть мати вкладену структуру. На кожному рівні її можна подати як сукупність рівневого центрального процесора і підпорядкованого йому спеціалізованого процесора, куди "вкладена" структура нижнього рівня. Незалежно від рівня центральний процесор взаємодіє із

спеціалізованим процесором через двопортову буферну пам’ять. Такий підхід сприяє реалізації системи на сучасних ПЛІС, для яких характерною є наявність у складі бібліотечних елементів і генерованих ядер вузлів двопортової пам’яті.

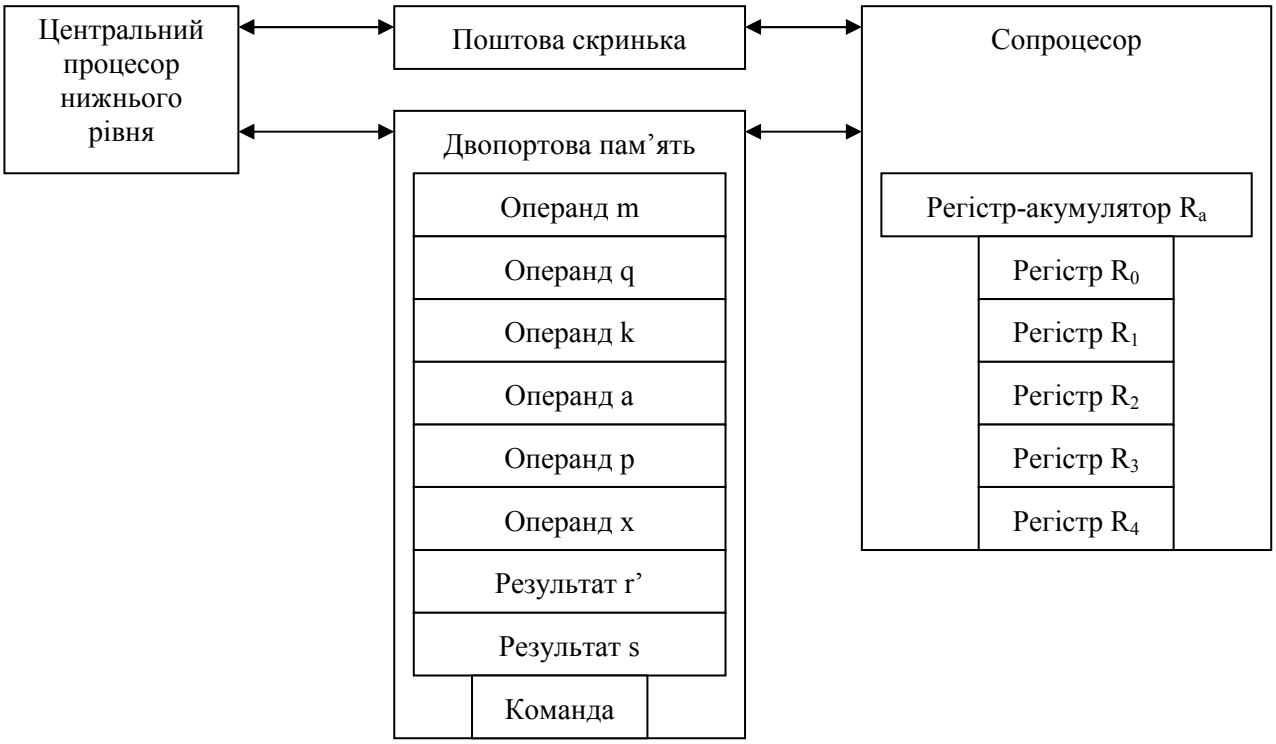

Рис. 2. Спецобчислювач

Для синхронізації роботи центрального та спеціалізованого процесорів використовується “поштова скринька” (“семафор”).

Незалежно від рівня центральний процесор при взаємодії із спеціалізованим виконує такі операції:

1) аналізує стан поштової скриньки і в разі відсутності там ознаки роботи спеціалізованого процесора переходить до виконання наступних операцій;
2) читає з двопортової пам’яті результат попередньої роботи спеціалізованого процесора;
3) заносить до двопортової пам’яті операнди;
4) записує до двопортової пам’яті команду;
5) встановлює у поштовій скриньці ознаку роботи спеціалізованого процесора (запускає його);
6) повертається на виконання першого кроку у даній послідовності операцій.

У свою чергу спеціалізований процесор виконує такі дії:

1) аналізує стан поштової скриньки і в разі наявності там ознаки роботи спеціалізованого процесора переходить до виконання наступних операцій;
2) читає з двопортової пам’яті операнди;
3) читає з двопортової пам’яті команду;
4) декодує і виконує команду;
5) записує до двопортової пам’яті результат;
6) скидає у поштовій скриньці ознаку роботи спеціалізованого процесора;
7) повертається на виконання першого кроку у даній послідовності операцій.

Особливості формування цифрового підпису. Аналіз алгоритмів формування і перевірки цифрового підпису показує, що їхньою особливістю є обробка багаторозрядних цілочисельних

даних. У табл. 1 наведено розрядності вхідних та вихідних даних і проміжних результатів, які потрібні для формування цифрового підпису згідно з ГОСТ 34.310–95 [2]. Послідовність виконання операцій над цими числами показана на блок-схемі алгоритму вироблення цифрового підпису, яка наведена на рис. 3.

Таблиця 1

Розрядність змінних

Змінна	Кількість біт	Вираз	Призначення
q	256		вхідна величина
p	512		вхідна величина
a	512	1<a<p-1	вхідна величина
h(M)	256	Функція хешування згідно з [13]	вхідна величина
x	256	0<x<q	вхідна величина – секретний ключ
k	256	0<k<q	вхідна випадкова величина
r	512	$a^k(\text{mod } p)$	проміжний результат
r’	256	r(mod q)	кінцевий результат
s	256	(xr’ + kh(M))(mod q)	кінцевий результат

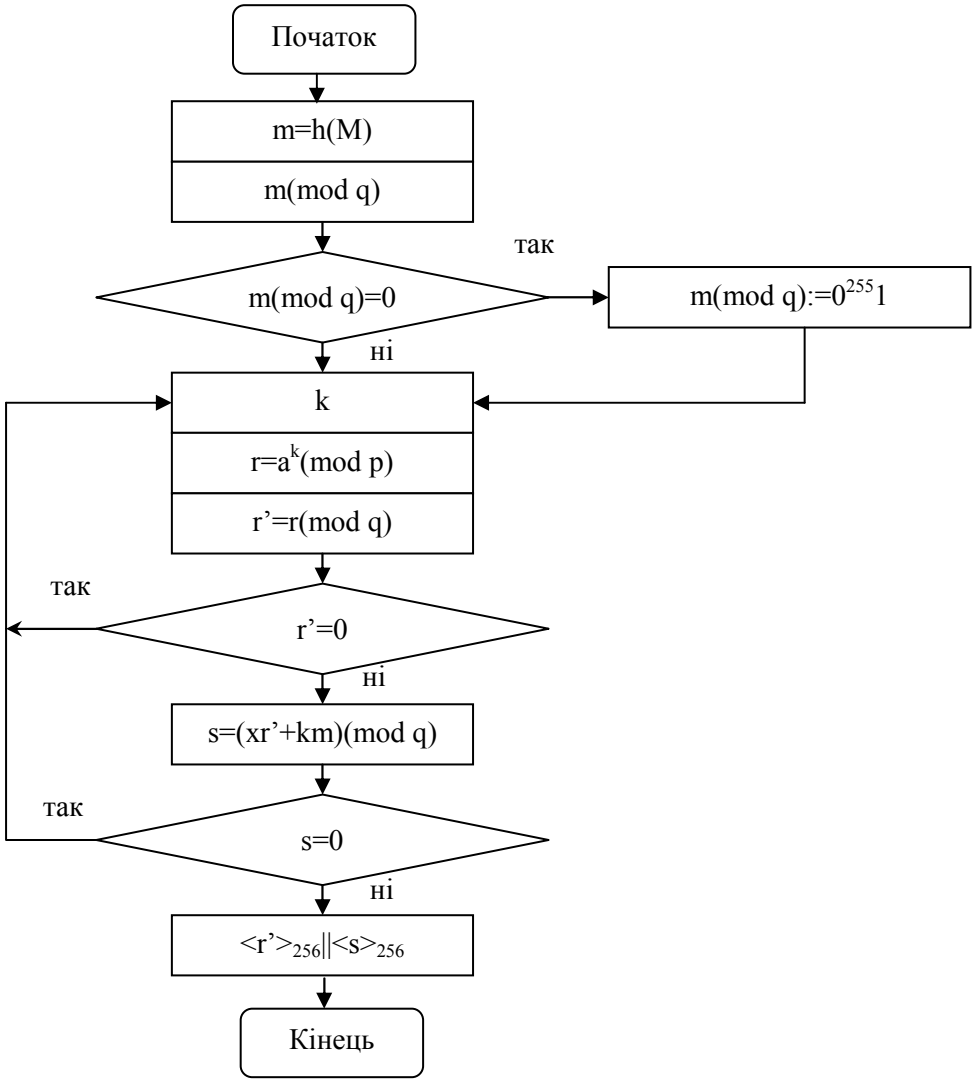

Рис. 3. Вироблення електронного цифрового підпису згідно з ГОСТ 34.310–95

З аналізу алгоритму випливає необхідність реалізації довгих операцій: піднесення до степеня за модулем і знаходження модуля (друга з них є окремим випадком першої).

Модуль (залишок від ділення) знаходиться при виконанні операції ділення одним з відомих способів, наприклад, методом без відновлення залишків.

Для знаходження степеня за модулем $a^k \pmod p$, відомий алгоритм [16]:

якщо $k = k_0 2^r + \ldots + k_{r-1} 2^1 + k_r 2^0$, $k_i = \{0;1\}$, $k_0 = 1$,

тоді $a_i = a_{i-1}^2 * a^{ki} \pmod p$, $a_0 = a$.

Іншими словами

$$a_i = a_{i-1}^2 * a \pmod p, \text{ якщо } k_i = 1.$$
$$a_i = a_{i-1}^2 \pmod p, \text{ якщо } k_i = 0. \quad (1)$$

Таким чином ця операція зводиться до ітераційного аналізу розрядів показника степеня k, виконання операцій множення і знаходження модуля.

Синтез системи команд криптографічного сопроцесора. З проведеного аналізу видно, що розрядність операндів і результатів кратна 256 біт. Тому для центрального процесора сопроцесор, який власне проводить обчислення, представляється як сукупність 256 розрядних регістрів, куди можна записати вхідні величини і звідки можна прочитати остаточні та проміжні результати. Для отримання цифрового підпису достатньо мати обмежену кількість таких регістрів, умовно позначених далі як регістри R_i (R_j) і один 512 розрядний регістр-акумулятор, далі позначений як регістр R_a.

На основі зроблених зауважень можна представити реалізацію алгоритму формування цифрового підпису як послідовність "канальних" команд спецобчислювача – його програму. Ця послідовність формується центральним процесором верхнього рівня. Кожна канальна команда відповідає одному оператору на блок-схемі алгоритму і розкладається у послідовність програм центрального процесора нижнього рівня, який за необхідності ініціює виконання сопроцесором окремих команд з системи команд останнього.

До основних програм центрального процесора нижнього рівня входять програми, наведені у табл. 2.

Таблиця 2

Основні програми

№ з/п	Позначення програми	Дія	Опис програми
1	Операнд -> R_a	завантаження регістра-акумулятора	операнд з центрального процесора (ЦП) записується в регістр-акумулятор сопроцесора
2	Операнд -> R_i	завантаження регістра	операнд з центрального процесора (ЦП) записується в один з регістрів сопроцесора;
3	R_i -> R_a	пересилання	пересилання вмісту одного з регістрів до регістра-акумулятора
4	R_a (rest) -> R_i	пересилання	пересилання залишку (модуля) до одного з регістрів сопроцесора
5	вивід R_i	вивід	вивід результату з регістра
6	R_a mod R_i->R_a (rest),z	знаходження залишку	ділення вмісту регістра-акумулятора сопроцесора ділиться на вміст одного з регістрів сопроцесора, залишок від ділення записується у молодші розряди регістра-акумулятра (rest), формується ознака рівності залишку 0 (z)
7	R_a+ R_i * R_j ->R_a	множення з накопиченням	добуток вмісту двох довільних регістрів сопроцесора додається до вмісту регістра-акумулятора

Блок-схему реалізації алгоритму формування цифрового підпису з використанням основних програм (табл. 2) наведено на рис. 4. В операторах блок-схеми підкреслені дії, які повинні виконуватися згідно з ГОСТ 34.310–95, а нижче показано, якими програмами ці дії реалізуються.

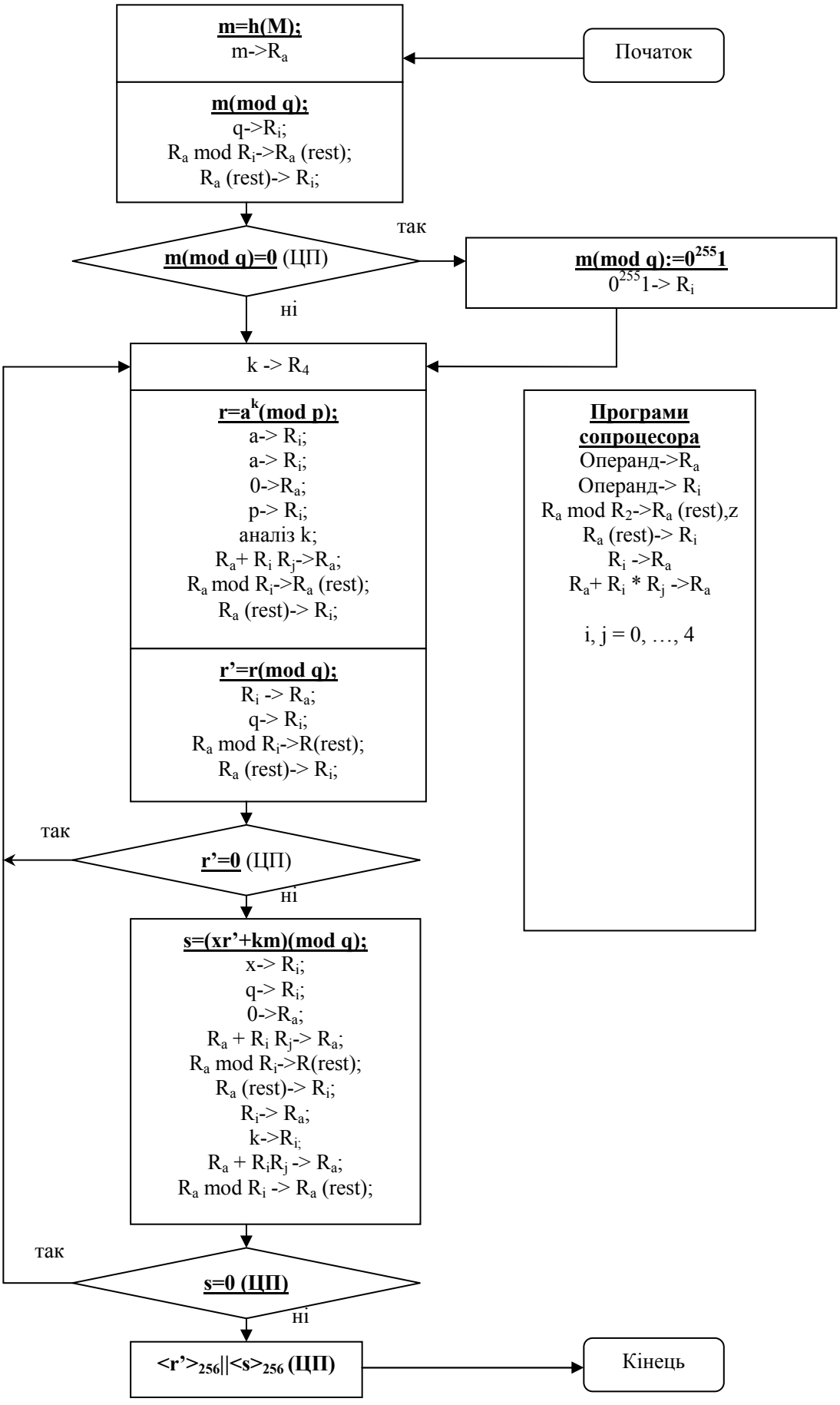

Рис. 4. Використання програм центрального процесора нижнього рівня

Основними програмами центрального процесора нижнього рівня є:

короткі програми (програми 1...5 табл. 2, кожній з цих програм відповідає одна команда сопроцесора);

програма множення з накопиченням (програма 7 табл. 2);

програма знаходження залишку (програма 6 табл. 2).

При знаходженні залишку треба виконувати значну кількість таких додаткових специфічних операцій, як:

визначення положення старших розрядів операндів та результатів. Для цього необхідно організувати зчитування окремих регістрів (де зберігається ділене і дільник) без видачі їхнього вмісту назовні сопроцесора;

визначення кількості циклів ділення;

вирівнювання положення старших бітів операндів;

визначення типу операції – додавання чи віднімання, яка повинна виконуватися в наступному циклі ділення (формування і аналіз знака проміжного результату).

Крім того, доцільно саму операцію ділення розглядати як послідовність коротких команд віднімання, додавання і зсуву (якщо ділення виконується методом без відновлення залишків). Тому доцільним є розбиття в першу чергу саме програми знаходження залишку на послідовність команд сопроцесора, які утворять основу його системи команд.

При знаходженні залишку виконується розподіл операцій між центральним процесором і сопроцесором. При цьому сопроцесор окремими командами виконує згадані вище додаткові специфічні операції, виконує додавання і віднімання з формуванням ознак нуля та знака, виконує зсув регістра-акумулятора, а центральний процесор визначає послідовність вказаних дій і залежно від ознаки знака під час віднімання і ділення визначає, яку з цих операцій треба виконувати сопроцесору в наступному циклі ділення.

Треба зауважити, що центральний процесор не тільки управляє роботою сопроцесора, але і сам може виконувати деякі прості операції. Наприклад, при піднесенні до степеня треба аналізувати розряди показника степеня. Оскільки показник степеня не модифікується під час знаходження залишку, доцільно аналіз його розрядів покласти на центральний процесор. При цьому сопроцесор виконує послідовність команд множення і додавання, а центральний процесор за результатами аналізу розрядів показника степеня визначає порядок цих операцій і кількість операцій множення в кожному циклі (1).

Також слід врахувати, що регістри сопроцесора і його регістр-акумулятор (де формується результат цілочисельного множення і знаходиться ділене при цілочисельному діленні) мають різну розрядність (відповідно, 256 і 512 біт). Тому для пересилання інформації між ними необхідно мати у складі команд сопроцесора команди пересилання 256 старших і 256 молодших бітів.

Результат синтезу системи команд сопроцесора наведений табл. 3. Виконання кожної команди сопроцесора ініціюється центральним процесором.

Таблиця 3

Система команд процесора

№	Позначення команди	Дії	Примітка
1	2	3	4
1	$R_a := R_a + R_a$	Зсув R_a на 1 біт ліворуч	Використовується під час виконання множення, під час підготовки до ділення і виконанні ділення
2	Read R_i		Визначення положення старшого розряду R_i (без виводу результату назовні, для перевірки проміжних результатів на етапі моделювання)
3	Read R_a		Визначення положення старшого розряду R_a (без виводу результату назовні, для перевірки проміжних результатів на етапі моделювання)

Продовження табл. 3

1	2	3	4
4	delta		Визначення кількості циклів ділення delta
5	$R_a := R_a - R_i$ 0…0, z,s	Віднімання	Старший ненульовий розряд повинен знаходитися в однакових позиціях, молодші розряди доповнюються нулями, використовується під час виконання ділення, формуються ознаки нуля і знака
6	$R_a := R_a + R_i$ 0…0, z,s	Додавання	Старший ненульовий розряд повинен знаходитися в однакових позиціях, молодші розряди доповнюються нулями, використовується під час ділення, формуються ознаки нуля і знака
7	R_a (lsb):= R_i	Пересилання	Пересилання у старші розряди регістра-акумулятора
8	R_a (msb):= R_i	Пересилання	Пересилання у молодші розряди регістра-акумулятора
9	$R_a := R_a + R_i * R_j$	Множення	
10…15			Команди, які відповідають коротким програмам 1…6 табл. 2

Для забезпечення відлагодження сопроцесора до складу його команд доцільно ввести технологічні команди, які забезпечують перевірку виконання тестових прикладів. Як тестові можуть використовуватися співвідношення 2*a+1 (mod a) =a+a+1 (mod a) = 1; 2*a – 1 (mod a) = a-1 та інші. Дані команди наведено у табл. 4 і у робочих програмах не використовуються.

Таблиця 4

Технологічні команди

№ з/п	Дії
1	Інкремент регістра-акумулятора (для перевірки правильності знаходження модуля)
2	Декремент регістра-акумулятора (для перевірки правильності знаходження модуля)

Висновки. Запропонований підхід до створення системи команд криптографічного сопроцесора базується на представленні про багаторівневу структуру криптографічної системи, що нагадує семирівневу модель взаємодії відкритих систем. Оператори криптографічного алгоритму утворюють систему команд спецобчислювача (систему команд канального рівня). Кожна команда спецобчислювача реалізується послідовністю команд сопроцесора, який входить до його складу (системою команд "фізичного" рівня). Послідовностями команд управляє центральний процесор відповідного рівня. Система команд сопроцесора обмежується арифметичними командами додавання, віднімання, множення з накопиченням, зсуву і пересилання, які виконуються над багаторозрядними цілими числами. Також до системи команд додаються технологічні команди, призначені для полегшення відлагодження сопроцесора.

Цей підхід ілюструє перехід від визначення алгоритму як припису до його визначення як "фіксованої для розв'язання деякого класу задач конфігурації апаратно-програмних засобів ..., що задає обчислювальний процес..." [17].

Запропонований підхід і система команд сопроцесора дозволяють скоротити час проектування і відлагодження сопроцесора за рахунок зменшення довжини послідовностей виконуваних ним операцій. Також полегшується процес розширення функціональних можливостей спецобчислювача.

Перевірка цього підходу і системи команд були зроблені під час реалізації спецобчислювача на ПЛІС xcv100-pq240-4 (Xilinx). Як центральний процесор використовувалося ядро PicoBlaze [18]. Апаратні витрати на реалізацію сопроцесора склали 439 слайсів (36 %), робота взірця була перевірена на відповідність ГОСТ 34.310–95.

1. Соболев О. Электронная цифровая подпись в Украине: началось внедрение ЭЦП // Чип – Украина № 11. 2003. С. 14–16. 2. Межгосударственный стандарт ГОСТ 34.310-95. Информационная технология. Криптографическая защита информации. Процедура выработки и проверки электронной

цифровой подписи на базе асимметричного криптографического алгоритма. Межгосударственный совет по стандартизации, метрологии и сертификации. Минск. Госстандарт Украины, с дополнениями, 1997. 3. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. – Львів: БаК, 2003. – 144 с. 4. Коркішко Т., Мельник А., Мельник В. Алгоритми та процеси симетричного блокового шифрування. – Львів: БаК, 2003. – 168 с. 5. Національний стандарт України ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що грунтується на еліптичних кривих. Формування та перевіряння. – К.: Держ. ком. України з питань технічного регулювання та споживчої політики, 2003. 6. Баричев С., Серов Р. Основы современной криптографии. – М.: "Горячая линия – Телеком", 2001. 7. Черемушкин А.В. Лекции по арифметическим алгоритмам в криптографии. – М.: Изд-во Моск. Центр. непрерывного математического образования, 2002. 8. Глухов В.С., Мельник А.О., Пуйда В.Я. Дослідження шляхів створення кодера та декодера відеосигналу // Вісн. Нац. ун-ту "Львівська політехніка". – 2003. – № 492. – С. 35–47. 9. Мельник А.О., Коркішко Т.А. Система підтримки виконання алгоритмів криптографічного захисту інформації на основі програмованого процесора та криптографічних акселераторів // Вісн. Держ. ун-ту "Львівська політехніка". – 2000. – № 385. – С. 77–80. 10. Коркішко Т.А., Мельник А.О. Вимоги до продуктивності процесів шифрування симетричними блоковими алгоритмами // Вісн. Нац. ун-ту "Львівська політехніка". – 2001. – № 437. – С. 83–90. 11. Морозов Ю.В. Інтелектуальна карта для системи цифрового підпису // Вісн. Нац. ун-ту "Львівська політехніка". – 2003. – № 492. – С. 107–111. 12. Попович Р.Б. Криптоаналіз системи RSA на основі пошуку функції Ейлера // Вісн. Нац. ун-ту "Львівська політехніка".– 2003. – № 492. – С. 128–133. 13. Аронов В.Б., Глухов В.С., Деревенко Я.К., Заиченко Н.В., Федуняк С.Ф. Одноплатный арифметический процессор, подключаемый к магистрали ГОСТ 26765.51-86, и средства обеспечения его серийного производства // 1-я научно-техническая конференция НПО "Фазотрон": Тез. докл. – М., 19–21 сентября 1989. 14. Глухов В.С., Заиченко Н.В. Арифметический спецвычислитель с кэш-памятью команд // Тез. докл. 29-й научн.-техн. конф. НПО "Антей". – М., 1990. 15. Межгосударственный стандарт ГОСТ 34.311-95. Информационная технология. Криптограмическая защита информации. Функция хэширования. Межгосударственный совет по стандартизации, метрологии и сертификации. – Минск: Госстандарт Украины, с доп., 1998. 16. Введение в криптографію / Под ред. В.В. Ященко. – М.: МЦНМО, 2000. 17. Черкаський М.В. Еволюція тлумачення поняття "алгоритм" // Вісн. Нац. ун-ту "Львівська політехніка". – 2003. – № 492. – С. 142–146. 17. Ken Chapman. PicoBlaze 8-Bit Microcontroller for Virtex-E and Spartan-II/IIE Devices. XAPP213 (Vol. 2.1) February 4, 2003. – Xilinx, Inc.