ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ

doi

УДК 004.274:004.056

С.Я. Гільгурт, канд. техн. наук
Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України
(Україна, 03164, Київ, вул. Генерала Наумова, 15,
тел. 066 756 43 48; e-mail: hilgurt@ukr.net)

Побудова асоціативної пам'яті на цифрових компараторах реконфігуровними засобами для вирішення задач інформаційної безпеки

Мережеві системи виявлення вторгнень, робота яких заснована на сигнатурному аналізі, виконують в реальному часі ресурсомістку задачу одночасного пошуку множини заданих рядків символів в інтенсивному потоці даних. Традиційні програмні рішення вже не задовольняють сучасним вимогам до їх швидкодії. Тому швидко набувають популярності апаратні прискорювачі на основі ПЛІС. Один з найпоширеніших підходів до побудови швидкодіючих схем розпізнавання на програмованій логіці заснований на застосуванні асоціативної пам'яті та цифрових компараторів, з яких вона складається. З метою підвищення ефективності створюваних реконфігуровних засобів інформаційної безпеки проаналізовано переваги та недоліки такого підходу, особливості його реалізації на ПЛІС, проблеми, що виникають, та шляхи їх вирішення.

Ключові слова: захист інформації, сигнатурний аналіз, ПЛІС, асоціативна пам'ять, цифровий компаратор, ефективність.

Проблеми інформаційної безпеки належать до категорії питань, що стоять перед людством, які навряд будуть принципово вирішені найближчим часом. Для їх подолання використовують різні заходи — організаційні, технічні, криптографічні тощо. Кожному напряму захисту притаманні свої складності та виклики. Мережеві системи виявлення вторгнень (МСВВ), антивіруси, засоби протидії мережевим хробакам та інші засоби технічного захисту, робота яких заснована на пошуку в інтенсивному потоці даних заздалегідь заданих ознак шкідливої активності (сигнатур), мають вирішувати складну задачу множинного розпізнавання рядків [1]. У зв'язку з припиненням зростання частоти традиційних процесорів, а також

через стале зростання злонавмисної активності в інформаційних системах програмні рішення вже не відповідають вимогам щодо швидкодії. Тому розробники почали частіше звертатись до апаратних платформ на основі програмованих логічних інтегральних схем (ПЛІС) [2]. Висока продуктивність в поєднанні з гнучкістю, близькою до програмної, якнайкраще відображає динамічну природу галузі інформаційного захисту. Але зростання складності завдань комп'ютерної безпеки триває, як і збільшення об'ємів мережевого трафіку. Тому підвищення ефективності реконфігуровних апаратних засобів захисту інформації є вельми актуальним завданням.

Для виконання цього завдання необхідно насамперед визначитися з показниками, за якими можна порівнювати окремі підходи, рішення та розроблені системи в цілому. Показники ефективності сигнатурних реконфігуровних засобів інформаційної безпеки можна умовно поділити [3] на вартісні, швидкісні (показники продуктивності) та функціональні.

До вартісних показників належать обсяги логічних ресурсів програмованої логіки, задіяні для створення цифрової схеми, витрати на пам'ять (як зовнішню відносно кристалу ПЛІС, так і внутрішню — блочну пам'ять BRAM та розподілену у вигляді тригерів логічних комірок), а також загальна вартість володіння, яка включає витрати на розробку, виготовлення, програмування та експлуатацію системи.

До параметрів продуктивності належать об'єм словника сигнатур, які розпізнає засіб, пропускна здатність, а також передбачуваність пропускної здатності.

До функціональних показників належать спроможність системи МСВВ працювати у режимі запобігання вторгнень, здатність до динамічного оновлення словника сигнатур без припинення процесу розпізнавання, здатність протидіяти атакам на МСВВ тощо.

Важливим проміжним показником, який пов'язує швидкісні характеристики з вартісними, є масштабованість, тобто здатність нарощувати продуктивні здібності без надмірних додаткових ресурсних витрат. Розрізняють масштабованість за пропускною здатністю, за об'ємом словника сигнатур та за довжиною патернів — послідовностей символів що підлягають пошуку.

У роботі [4] подано ідею методу створення універсальної комбінованої структури модуля розпізнавання реконфігуровної системи захисту інформації. Її суть полягає у поєднанні (з метою підвищення ефективності) в одному пристрої переваг різних підходів до побудови схем розпізнавання на основі ПЛІС, яких у світі на даний час створено чимало. Для ефективної реалізації методу вирішальним чинником є глибоке розуміння технічних властивостей кожного з підходів.

При створенні засобів інформаційного захисту на основі ПЛІС найкращими виявилися підходи [4], засновані на використанні асоціативної пам'яті (АП) та цифрових компараторів (ЦК), хеш-функцій та цифрових автоматів.

Дане дослідження присвячено всебічному аналізу першого з перелічених підходів. При цьому використано багатий досвід, здобутий чисельними розробниками програмованих засобів інформаційного захисту зі всього світу. В процесі вивчання предмету дослідження головну увагу було спрямовано на таке:

особливості (переваги та недоліки) підходу, що вивчається, в сенсі забезпечення показників ефективності, сформульованих вище;

специфіку реалізації підходу на ПЛІС;

складнощі та проблеми, що виникають під час створення реконфігуровних засобів, і шляхи їх подолання з метою покращення показників ефективності.

Асоціативна пам'ять на ЦК. Асоціативна пам'ять (Content Addressable Memory (CAM)) — це клас пристроїв, що від самого початку створювалися для швидкого розпізнавання патернів [5, 6]. В одному режимі вони працюють як стандартні схеми пам'яті довільного доступу (Random Access Memory (RAM)) і можуть використовуватися для зберігання бінарних даних. Але на відміну від останніх, АП здатна забезпечити також потужний паралельний режим розпізнавання. Цей режим дозволяє шукати всі дані в пристрої одночасно, за один цикл тактової частоти [7]. При цьому АП фактично виконує функцію, протилежну традиційному оперативному запам'ятовуючому пристрою (ОЗП). Якщо ОЗП, приймаючи на вхід адресну інформацію, видає зміст відповідної комірки пам'яті, то АП навпаки, за змістом відшукує його місце розташування або сигналізує про відсутність питомих даних у пристрої.

Протягом десятиліть АП використовувалася в обчислювальній техніці переважно для керування процесом доступу до кеш-пам'яті. При цьому ключовим питанням, яке потрібує якомога скорішого вирішення, є визначення, чи відбувся «хіт» або «промах». Саме здатність АП до паралельного апаратного пошуку обумовлює її використання для керування кеш-пам'яттю, де критичне значення має швидкодія. Але згодом АП як апаратне рішення почали все частіше застосовувати в мережевих технологіях [8]. Обробка пакетів даних, що надходять у мережевий маршрутизатор, залежить від адреси призначення мережі пакета. Через велику кількість потенційних адрес і зростаючі вимоги до продуктивності АП почали поширювати на обробку інформації про мережеві адреси. Найновітніша сфера застосування АП пов'язана з поглибленим

аналізом мережевих пакетів з метою викриття зловмисної активності у системах захисту інформації.

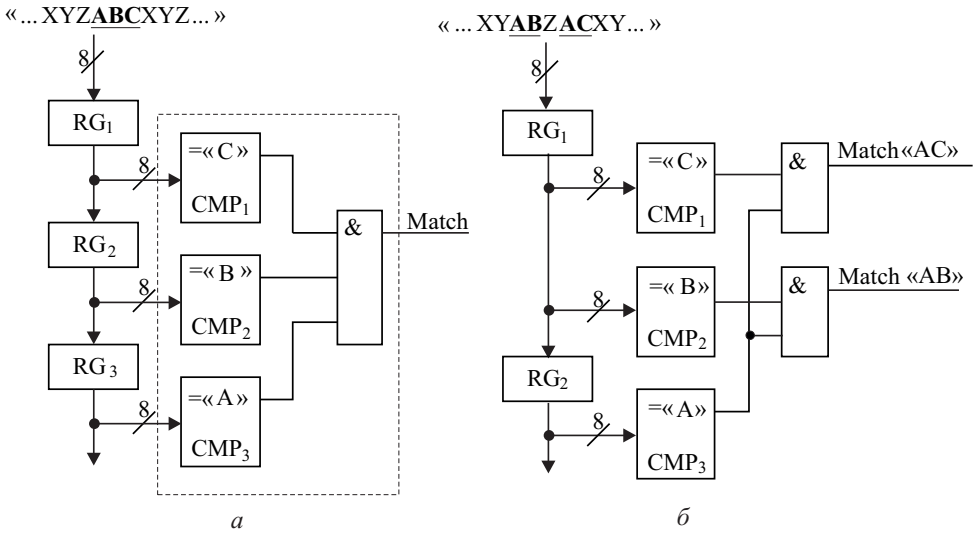
Використання АП у мережевих додатках призвело до підвищення інтересу до цієї ніші виробників комерційних пристроїв. В результаті на ринку з'явилися апаратні спеціалізовані мікросхеми АП. Наприклад, компанії NetLogic [9] та Music Semiconductor [10] почали виробництво подібних мікросхем спеціально для мережевого ринку [11].

На жаль, окрім переваг спеціалізовані мікросхеми АП мають ті ж самі недоліки, що й будь-які апаратні вироби з фіксованою структурою. Вони забезпечують максимальну швидкодію, але жорстка архітектура обмежує гнучкість використання. Частково подолати це обмеження дозволяє такий різновид АП як трійкова АП (ТАП) (Ternary Content Addressable Memory (TCAM)). На відміну від традиційної, бінарної АП, у ТАП до значень «0» та «1», які може приймати кожен біт, додається третє значення для порівняння, яке має сенс «не важливо» [12, 13]. Таке розширення надає додаткову гнучкість пошуку, зокрема спрощує розпізнавання на одному й тому ж апаратному пристрої з фіксованою організацією пам'яті патерни з різною довжиною, а також дозволяє виконувати найпростіші варіанти гнучкого пошуку [14].

У роботі [12] розглянуто розробку сигнатурної апаратної системи розпізнавання на основі мікросхеми промислової ТАП типу 5512GLQ ємністю 240 Kb виробництва фірми NetLogic Microsystems. Рішення дозволяє розпізнавати 1836 патернів довжиною від 10 до 100 символів бази сигнатур мережевої МСВВ Snort [15] або1768 патернів довжиною від 6 до 2189 символів зі словника антивірусної системи ClamAV [16], забезпечуючи пропускну здатність у 2 Гб/с. При цьому реалізовано такі види гнучкого пошуку, як зв'язані патерни та операція заперечення.

Крім обмежень жорсткої логіки, спеціалізовані мікросхеми АП потерпають від відсутності зручної платформи для практичного використання, яку мають ПЛІС у вигляді стандартизованих прискорювачів [17]. Частково подолати цю проблему дозволили так звані мережеві процесори (Network Processors). Відомо кілька розробок для сигнатурних систем інформаційного захисту, виконаних на основі мережевих процесорів, до складу яких входять спеціалізовані мікросхеми АП [18—20]. Але в останні роки інтерес розробників засобів інформаційного захисту до пристроїв АП з фіксованою логікою зменшився. Отже, найкращою базою для побудови АП в складі систем інформаційної безпеки виявляється програмована логіка ПЛІС.

Оскільки швидкодіючою основою АП, що забезпечує її головні властивості, є ЦК, будемо використовувати ці два поняття як синоніми. Під



а *б*

Рис. 1. Схеми розпізнавання компараторами (*а*) та сумісного їх використання (*б*)

сигнатурою будемо розуміти сукупність відомостей щодо конкретної атаки або загрози. Патерн — найтиповіша складова сигнатури, яка містить фіксовану кількість окремих символів, поданих у вигляді байтів. Код символу може належати до всього діапазону можливих двійкових комбінацій з восьми бітів. Задача розпізнавання патерну полягає у пошуку відповіді на питання: чи присутній у вхідній послідовності символів фрагмент, який повністю збігається з будь-яким з патернів бази даних сигнатур.

Базова схема розпізнавання патернів на ЦК. Безпосередньою реалізацією функції виявлення збігу вхідного слова з патерном є набір дискретних компараторів, кожен з яких порівнює вхідний байт із заздалегідь заданими символами патерну [21, 22]. На рис. 1, *а*, наведено схему, яка містить конвеєр з восьмирозрядних регістрів RG_i, компаратори CMP_1 … CMP_3, кожен з яких виконує функцію порівняння з певним символом, та логічний елемент «І», що об'єднує їх виходи. Набір компараторів відповідає одному з патернів, які підлягають розпізнаванню. На вхід конвеєру подається послідовність символів, що аналізується. В разі збігу фрагменту послідовності з патерном «ABC» на виході Match з'являється активний сигнал. З метою уникнення подальших непорозумінь слід звернути увагу на те, що перший символ патерну відповідає останньому ступеню конвеєра.

Найважливіша перевага цієї схеми — висока швидкодія. У граничному випадку результат на виході з'являється за один такт синхронізації після подачі на вхід шуканої послідовності символів. До переваг схеми слід також віднести регулярність структури та простоту її синтезу.

Найсуттєвішим недоліком даної схеми порівняно з іншими підходами є значне споживання ресурсів, а саме логіки та розподіленої пам'яті (тригерів логічних комірок ПЛІС). Недоліком також є високе енергоспоживання та погана масштабованість за об'ємом словника сигнатур.

Розглянемо причини вказаних проблем та шляхи їх подолання.

Особливості реалізації схеми ЦК на ПЛІС. Схема на рис. 1, *а*, виглядає досить простою. Для того щоб краще зрозуміти труднощі її практичного застосування реконфігуровними засобами, звернемо увагу на такий факт. Вихідні лінії регістра RG_1 подаються на компаратори, що відповідають останньому символу кожного з патернів словника сигнатур, виходи регістра RG_2 — на компаратори передостаннього символу кожного з патернів і так далі. Оскільки словник сигнатур сучасних МСВВ містить велику кількість патернів, здатність навантаження (fan-out) виходів цих регістрів, синтезованих на штатних компонентах ПЛІС, є недостатньою. В той же час, довжина цифрових ліній, що зв'язують велику кількість логічних елементів (ЛЕ), розподілених по площині кристалу ПЛІС, зумовлює затримки розповсюдження сигналу, внаслідок чого знижується максимально можлива частота функціонування схеми. Для вирішення цієї проблеми створюють конвеєр з декількох ступенів, на кожному з котрих вихідні сигнали розгалужуються на множину входів D-тригерів наступного ступеня [23]. Таке рішення дозволяє розподілити виходи регістрів RG_i на довільну кількість компараторів без зниження тактової частоти.

Інша проблема пов'язана з надвеликою кількістю входів ЛЕ «І» схеми на компараторах, оскільки довжина патернів може сягати десятків символів, а пошукові таблиці LUT сучасних ПЛІС, на яких реалізуються ці елементи, мають від чотирьох до восьми входів. Ця проблема також вирішується за допомогою конвеєризації [21, 23].

У результаті аналізу виникаючих складностей можна побачити, що при збільшенні кількості патернів, які мають бути розпізнані, розглянуті вище проблеми, по-перше, спричиняють нелінійне зростання апаратних витрат, тобто обумовлюють погану масштабованість підходу за об'ємом словника сигнатур. По-друге, конвеєризація збільшує затримку поширення сигналу вздовж цифрової схеми на стільки тактів, скільки ступенів містить конвеєр. Тобто гарні показники швидкодії, які є головною перевагою схем АП, також погіршуються із збільшенням числа патернів. По-третє, конвеєризація загострює проблему високого енергоспоживання базової схеми на ЦК.

Тому розробниками було запропоновано декілька прийомів та технічних рішень, які дозволили суттєво знизити ресурсоємність та покращити інші показники ефективності даного напрямку побудови реконфігуровних засобів розпізнавання.

Зменшення ресурсних витрат. Головна ідея скорочення витрат на апаратні ресурси АП полягає у повторному використанні компараторів для розпізнавання різних патернів. У схемі, поданій на рис. 1, *б*, компаратор CMP_3 задіяний для розпізнавання одночасно двох патернів: «AB» та «AC» [24]. Подальший розвиток ідеї повторного використання приводить до граничного рішення, у якому для кожного символу словника сигнатур використовується лише один компаратор, а всі комбінації, потрібні для розпізнавання кожного з патернів, формуються за допомогою відповідних цифрових схем затримки (рис. 2). Таке рішення отримало назву Decoded Content-Addressable Memory (DCAM), зумовлену тим, що повний комплект, який налічує 256 компараторів для кожної можливої комбінації бітів у вхідному байті, фактично являє собою дешифратор (decoder) 8/256 [24].

Очевидними перевагами рішення DCAM є наступні:

скорочення числа восьмирозрядних компараторів від величини сумарної кількості символів у словнику сигнатур (у базовій схемі) до 256 штук (у схемі DCAM);

відпадає потреба у восьмирозрядному конвеєрі довжиною, що дорівнює розміру найдовшого патерну в словнику.

Недоліком DCAM виявляється необхідність синтезу великої кількості схем затримки, яка в загальному випадку також повинна дорівнювати сумарній кількості символів у словнику сигнатур. Але це число можна скоротити через повторне використання, зокрема для однакових символів, які займають ту ж саму позицію у різних патернах (наприклад, затримка DEL1 в схемі на рис. 2, *а*).

Фактором, що сприяє використанню техніки DCAM, є те, що для більшості сучасних ПЛІС існують бібліотечні елементи, які надають можливість економно синтезувати цифрові схеми затримки. Наприклад, в системах САПР фірми Xilinx існує типовий компонент зсувного регістру SRL16, який дозволяє, використовуючи чотирьохрозрядну пошукову таблицю LUT логічної комірки, створити схему затримки на довільну величину від одного до 16 тактів [25].

Зауважимо, що використання техніки повного дешифрування вхідних символів у однодротові лінії використовується не лише в схемах АП на ЦК [26, 27].

Подальшим напрямом зменшення складності схеми розпізнавання на базі DCAM є використання техніки часткового розпізнавання, суть якої полягає у наступному. Для зменшення витрат апаратних ресурсів схемою DCAM розбиваємо довгі патерни на коротші підрядки та послідовно розпізнаємо кожний підрядок. При цьому не потрібно затримувати дані з дешифратора на велику кількість тактів. Натомість достатньо лише затримувати сигнал часткового збігу.

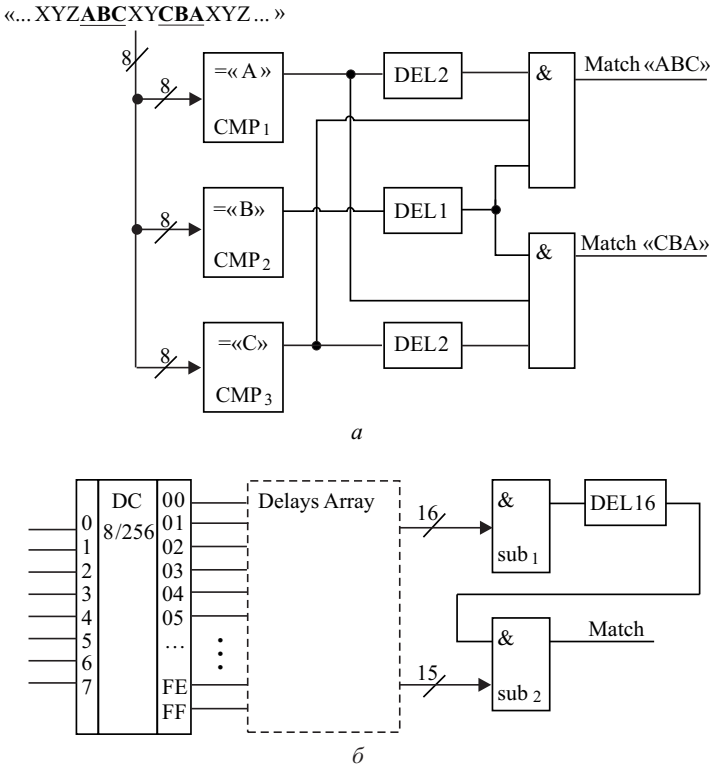


Рис. 2. Схемне рішення DCAM (*а*) і DpCAM (*б*)

На рис. 2, *б*, наведено схему розпізнавання 31-символьного патерну [28]. Послідовність символів, що аналізуються, подається на вхід дешифратора DC. Перетворені на однодротові лінії символи поступають на масив цифрових схем затримки Delays Array. Лінії, що відповідають першим 16 символам питомого патерну з виходів відповідних схем затримки подаються на ЛЕ sub_1. Лінії, що відповідають останнім 15 символам питомого патерну, з виходів відповідних схем затримки подаються на ЛЕ sub_2. На цей же елемент подається сигнал з виходу елемента UP після затримки на 16 тактів схемою DEL16. Кожен з ЛЕ «І» на 16 входів може бути синтезований на п’ятьох чотирьохвходових пошукових таблицях або на трьох восьмивходових.

Таке рішення в роботі [28] отримало назву Decoded partial Content-Addressable Memory (DpCAM). За допомогою додаткової лінії затримки на 16 тактів схема дозволяє зекономити 15 ліній затримки довжиною від 17 до 31 такту. В загальному випадку розпізнавання патерну довжиною L можна здійснити шляхом K-кратного каскадування по S символів, де $K = \lfloor L / S \rfloor$. Для цього потрібно мати додатково K схем затримки на S тактів замість

звільнених схем затримки кількістю від одного до $L - KS$ та довжиною від $S + 1$ до L тактів. До недоліків техніки DpCAM можна віднести залежність її ефективності від складу та властивостей словника сигнатур, що зумовлює регулярність структури.

Розпаралелювання. Як показано вище, АП на ЦК в якості засобу розпізнавання притаманна погана масштабованість за об'ємом словника сигнатур. Розглянемо, як масштабується цей підхід за пропускною здатністю. Для прискорення роботи базової схеми розпізнавання (див. рис. 1, *а*) в m разів за допомогою розпаралелювання на m потоків необхідно об'єднати m таких схем та подати на кожну з них вхідну послідовність зі зсувом на один символ.

На рис. 3, *а*, наведено приклад розпаралелювання для $m = 4$ [21]. Тут кожен блок SCH_i містить схему, означену штриховою лінією на рис. 1, *а*. Темп пересування інформації у конвеєрі в чотири рази вищий за швидкодію блоків розпізнавання. Тобто вхідна послідовність просувається на чотири символи вперед на кожному циклі обробки інформації блоками SCH_i. Як бачимо, m-кратне прискорення приводить до m-кратного збільшення апаратних ресурсів при несуттєвих додаткових витратах. Тобто розпаралелювання базової схеми на цифрових компараторах має добру масштабованість за пропускною здатністю.

Результати подальшого дослідження свідчать про те, що використання описаних вище технік зменшення витрат у компараторних схемах розпізнавання при їх розпаралелюванні приводить до певної економії ресурсів порівняно з лінійним зростанням.

Прототипом схеми розпаралелювання з використанням техніки DCAM для $m = 2$ (див. рис. 3, *б*) є Fig. 5 у роботі [24]. Вхідна послідовність просувається на два символи вперед на кожному циклі розпізнавання. Вихід Match сигналізує про збіг з патерном. Оскільки заздалегідь невідомо, з яким зсувом з'явиться питома комбінація символів, парним чи непарним, потрібні додаткові виходи xABC та ABCx, які дозволяють це з'ясувати. Як бачимо, при довжині патерна в три символи у схемі присутні всього три лінії затримки по одному такту. При цьому у відповідній однопотоковій схемі (див. рис. 2, *а*, вихід Match_ABC), яка вдвічі повільніша, задіяна одна затримка на один такт и одна — на два такти, тобто фактично така сама кількість ресурсів. Цей ефект зменшення кількості та довжин ліній затримки обумовлено тим, що вхідна послідовність також зсувається між тактами розпізнавання. Отже, схема двократного прискорення (див. рис. 3, *б*) збільшує потрібні ресурси менше, ніж удвічі, тобто за пропускною здатністю вона масштабується за сублінійним законом.

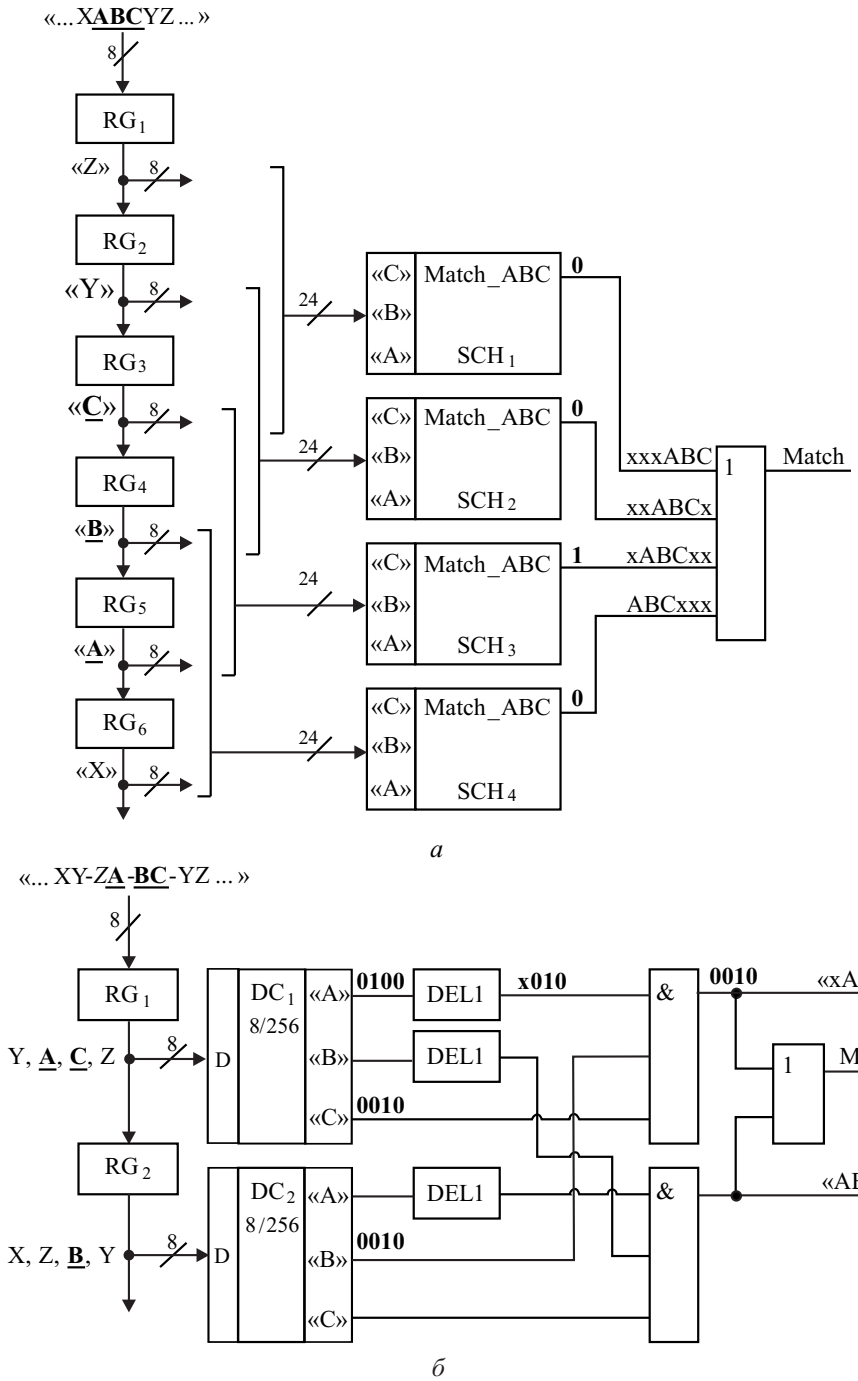


Рис. 3. Схеми розпаралелювання: *а* — чотирьохпотокового; *б* — двохпотокового з використанням техніки DCAM

Небайтова розрядність обробки даних. Ще одним напрямком досліджень стосовно зменшення апаратних ресурсів при синтезі блоків розпізнавання взагалі та при побудові схем на ЦК зокрема є відхід від побайтового принципу обробки інформації. В роботі [23] висловлено та досліджено ідею, базовану на використанні так званих півбайтових компараторів (Half-Byte Comparators (HBC)). При цьому кожен символ розглядається як складений з двох півбайтів — старшого та молодшого.

На рис. 4 надано схему розпізнавання патерну «ABCA», в якій одночасно використано дві техніки: DCAM та HBC. Старший (розряди з сьомого по четвертий) та молодший (розряди з третього по нульовий) півбайти вхідної послідовності подаються відповідно на компаратор HBC_1 та на компаратори HBC_2 — HBC_4, виходи яких після потрібних затримок об'єднуються схемою «І». Як бачимо, замість трьох повнобайтових компараторів в схемі використано всього чотири півбайтові компаратори. Але головною перевагою техніки виявляється той факт, що замість одного дешифратора 8/256 для розпізнавання всіх чотирьохбітних комбінацій півбайтів достатньо двох дешифраторів організацією 4/16. Тобто на вхід масиву цифрових ліній затримки подаються всього 32 дроти замість 256.

Зменшенню задіяних ресурсів сприяє також той факт, що символами, які входять до складу патернів словників сигнатур систем виявлення вторгнень, у більшості випадків є літери англійського алфавіту. Це означає, що для половини алфавіту (окремо для малих та великих літер) старші півбайти мають те саме значення (4_{16} — для літер від «A» до «P», 5_{16} — для літер від «Q» до «Z», 6_{16} — для літер від «a» до «p» та 7_{16} —для літер від «q» до «z»). Отже, велику кількість цифрових схем затримки можна зекономити в результаті їх сумісного використання.

До переваг схеми слід також віднести можливість синтезувати півбайтові компаратори на одній пошуковій таблиці LUT, навіть на чотиривходовій. Можна перевірити та засвідчитися в тому, що техніка використання півбайтових компараторів цілком сумісна з технікою розпаралелювання для прискорення розпізнавання. При цьому ефект зменшення кількості та довжини ліній затримки також спостерігається.

На рис. 5 наведено схему розпізнавання патерну «ABC» з двопотоковим розпаралелюванням, у якій використано півбайтові компаратори. Як і в схемі на рис. 3, *б*, вхідна послідовність тут просувається на два символи вперед на кожному циклі розпізнавання. При цьому апаратні ресурси збільшені менше, чим вдвічі. Якщо дешифраторів 4/16 потрібно рівно два комплекти, то ліній затримки на один такт схема потребує також п'ять одиниць, як і базова схема на рис. 4, яка вдвічі повільніша. Тобто паралельна схема на півбайтових компараторах масштабується стосовно швидкодії також за сублінійним законом.

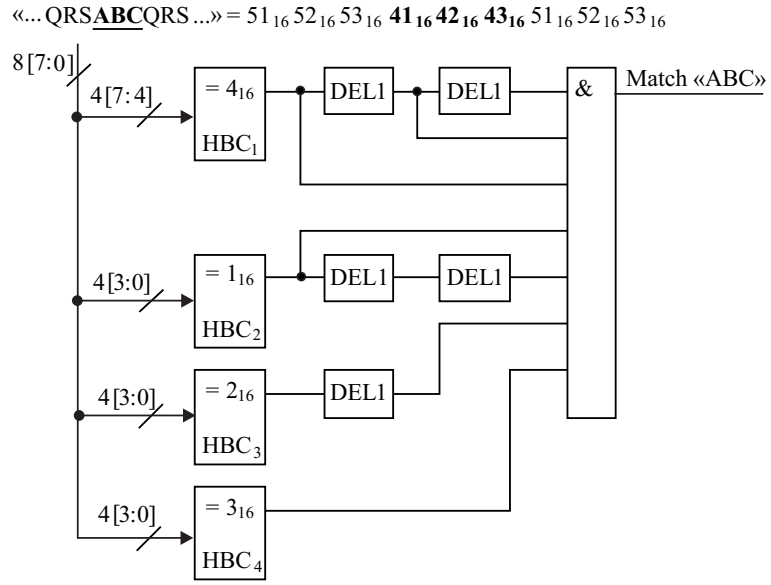


Рис. 4. Схема використання півбайтових компараторів

Інформація, що обробляється в схемі розпізнавання, може бути згрупована не тільки по вісім або по чотири біти. Кількість двійкових розрядів, що утворюють елементарну частку інформації, в граничному випадку може дорівнювати одиниці. Тобто елементарні символи, з яких складаються патерни, можуть кодуватися одним бітом. В роботі [29] запропоновано реалізацію на ПЛІС схеми розпізнавання, принцип дії якої ґрунтується саме на бітовому рівні. В її основу закладено математичний апарат бінарних діаграм рішень (БДР) (Binary Decision Diagram (BDD)).

Бінарна діаграма рішень — це механізм подання булевої функції n змінних $f(x_1, x_2, ..., x_n)$ у вигляді орієнтованого ациклічного графа, який складається з декількох вузлів рішень і термінальних вузлів двох типів (0-термінал і 1-термінал). Кожен внутрішній вузол рішень на i-му рівні позначений булевою змінною x_i та має по два нащадки — молодший та старший. Перехід від внутрішнього вузла x_i до молодшого або старшого нащадка виконується в залежності від поточного значення змінної x_i відповідно 0 або 1. Якщо для заданого набору значень $x_1, x_2, ..., x_n$ шлях від кореневого вузла приводить до 1-терміналу або 0-терміналу, це означає, що функція f для цього набору змінних дорівнює відповідно одиниці або нулю [30]. На рис. 6 наведено приклад БДР для булевої функції $f = (x_1 \vee x_2) \wedge x_3$ [31].

Однак більш зручною для застосування є так звана скорочена впорядкована бінарна діаграма рішень (СВБДР), яка здобувається зі звичайної

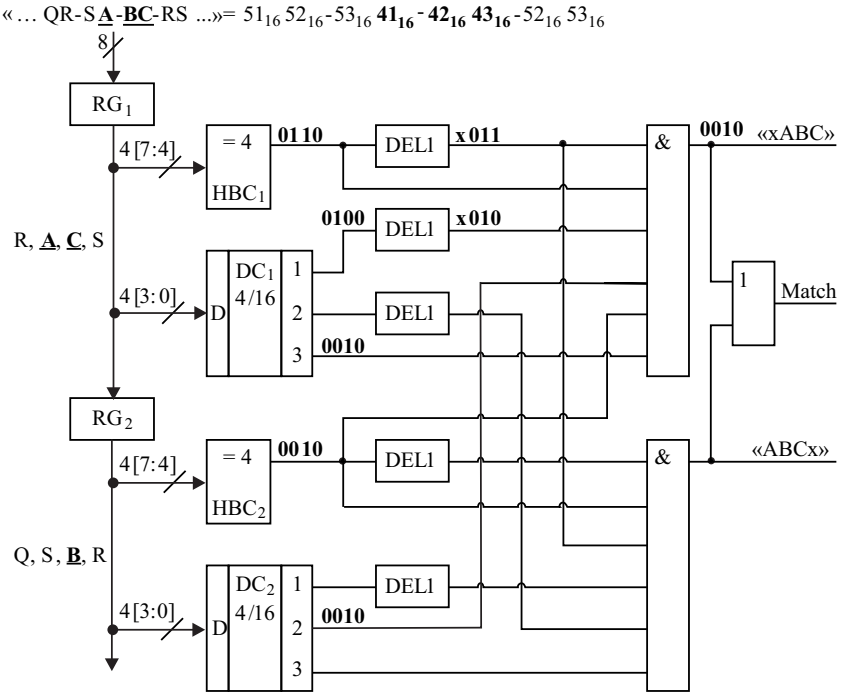


Рис. 5. Схема двопотокового розпаралелювання САМ на півбайтових компараторах

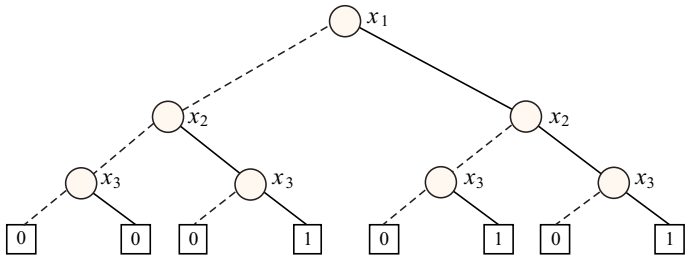


Рис. 6. Схема БДР для $f=(x_1 \vee x_2) \wedge x_3$ [31]

впорядкованої БДР в результаті злиття ізоморфних підграфів та вилучення вузлів, у яких обидва нащадки ізоморфні. Користь скороченої БДР полягає в тому, що вона однозначно представляє певну функцію для заданого порядку змінних. СВБДР має тільки два термінальних вузли: один 1-термінал та один 0-термінал. На рис. 7 надано схему СВБДР для БДР, наведеної на рис. 6. Як бачимо, СВБДР дозволяє подати БДР булевої функції в однозначному та компактному вигляді. Якщо уявити схему розпізнавання у вигляді чорної скриньки, на вхід якої побітово подається фрагмент вхідної послідовності символів, а вихідна бітова змінна приймає значення 1 або 0 в

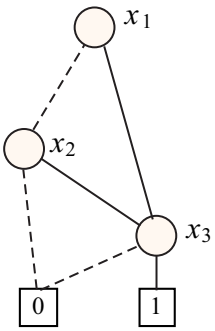


Рис. 7. Скорочена впорядкована БДР

залежності від того, чи збігається вхідний рядок з будь-яким з патернів, закладених в булеву функцію чорної скриньки, то БДР цієї функції відобразить логіку роботи такої схеми на бітовому рівні.

У роботі [29] запропоновано п'ятикрокову методику побудови пристроїв АП на основі БДР (BDD-based CAM (BCAM)) для реалізації на ПЛІС. В процесі виконання методики з вхідної бази даних сигнатур (наприклад, MCBB Snort) обирають патерни, для кожного символу яких створюють БДР, які потім зливають в єдину БДР рядка. Далі операція повторюється для всіх патернів словника. Надлишковість множини патернів призводить до того, що окремі гілки діаграми дублюють одна одну. Автори методики використовують цей ефект наступним чином. Вони виокремлюють одну так звану основну БДР, яка об'єднує спільні для всіх патернів біти, а також низку неосновних БДР, які відображають відмінності кожного патерну. В загальному випадку кількість неосновних БДР може бути менша за кількість патернів. Далі по здобутим бінарним діаграмам рішень будують компаратори. Для основної БДР компаратор не відрізняється від звичайних, які використовуються в АП. По неосновним діаграмам компаратори будують, використовуючи пошукові таблиці LUT як мультіплексори.

На рис. 8 наведено схему розпізнавання с використанням механізму БДР [29]. Як приклад обрано чотири патерни довжиною по чотири символи (32 біти). Вхідну послідовність подано бітовими сигналами, з яких на входи компараторів (одного основного Com. CMP та чотирьох неосновних N/com. CMP_i) подається потрібна кількість бітів. Виходи неосновних компараторів, підтверджені сигналом з основного, свідчать про присутність у вхідній послідовності відповідного питомого рядка.

На наступному кроці неосновні компаратори модифікуються так само, тобто кожен з них перетворюється в основний та кілька неосновних компараторів. Процедура повторюється рекурсивно до повної оптимізації схеми. В результаті з початкової БДР утворюється компактна та ефективна апаратна схема BCAM.

За результатами практичної реалізації та тестових випробувань автори розробки стверджують, що оптимізована на бітовому рівні схема, побудована з використанням розробленої методики, потребує в два рази менше ресурсів ПЛІС порівняно з розпаралеленою DCAM [24].

До недоліків розглянутої техніки слід віднести її специфічність і несумісність з іншими техніками побудови схем на основі ЦК, зокрема DCAM, а також складність розпізнавання патернів різної довжини.

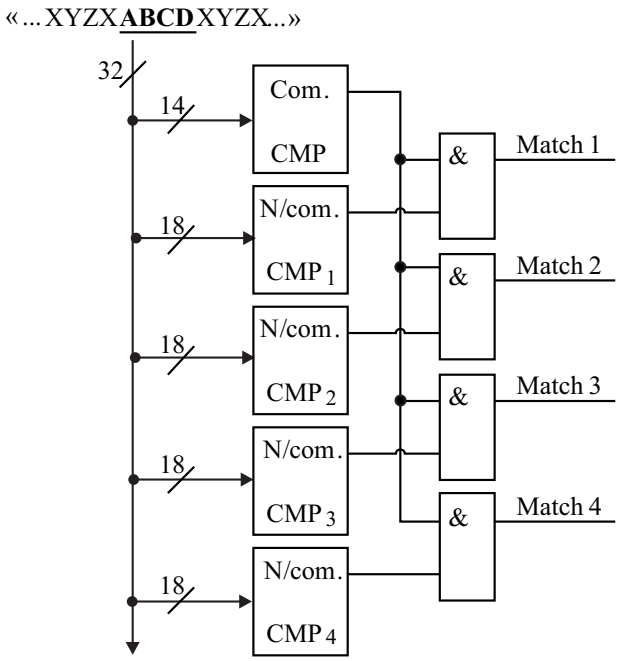


Рис. 8. Схема розпізнавання, побудована на основі БДР

Кластеризація. Розглянуті вище техніки дозволяють покращити показники швидкодії та економії ресурсів порівняно з базовою схемою. Але при їх практичній реалізації одна з головних негативних рис АП — погана масштабованість за об'ємом словника сигнатур — в деяких випадках виявляється критичною. У роботі [24] наведено приклад практичної розробки модуля розпізнавання для МССВ на основі ЦК, коли множина патернів містить загалом близько 100 000 символів, закодованих 8-бітним кодом. При цьому із всіх можливих 255 комбінацій бітів було задіяно лише 250. Здатність потрібного навантаження виходів дешифратора досягла значення 40, а довжина ліній з'єднань виявилася непридатно великою.

Єдиним рішенням, яке здатне принципово здолати проблему поганої масштабованості за об'ємом словника сигнатур, виявляється кластеризація, тобто розділення множини патернів на дрібніші групи. Тоді єдина велика схема розпізнавання розпадається на кілька підсхем меншого розміру, і на вхід кожної з них одночасно подаються питомі символи. В результаті, по-перше, внаслідок зменшення розміру підсхем послабляється вимога до здатності навантаження на виходи дешифраторів, по-друге, кожна окрема підсхема займає меншу площину на кристалі ПЛІС, що спричиняє скорочення довжини провідникових з'єднань [24].

Недоліком такої техніки є необхідність дублювати дешифратори, тобто синтезувати для кожної підгрупи окремий перетворювач вхідних символів в унітарний код. Частково пом'якшити ситуацію дозволяє використання такого алгоритму кластеризації, який зумовлює скорочення алфавіту підмножин патернів у підсхемах, що зменшує витрати на дешифратори. Найпростішою реалізацією подібного алгоритму виявилося звичайне сортування патернів. Запропонований у роботі [24] складніший алгоритм не надає суттєвих переваг ні в економії ресурсів, ні у швидкодії.

Підтвердженням ефективності техніки кластеризації можна вважати отриманий у роботі [24] результат, згідно з яким множину патернів розміром приблизно 100000 символів вдалося розбити на 24 групи в середньому по 54 символи в кожній, внаслідок чого розрядність дешифраторів в підсхемах скоротилася до шести. Слід зазначити, що розглянута техніка кластеризації не є єдиним варіантом розбивки набору патернів на підгрупи. Але дослідження інших способів виходить за рамки даної роботи.

Можливість динамічної реконфігурації. Як свідчать результати проведеного аналізу, при створенні засобів розпізнавання на ЦК інформація з бази даних сигнатур фактично «прошивається» у апаратну схему на самому нижчому рівні. З цього випливає ще один важливий недолік АП — занадто висока складність, тобто практична неможливість реалізації режиму динамічної реконфігурації.

У роботі [11] описано метод синтезу ЦК на пошукових таблицях LUT замість тригерів. Метод дозволяє, використовуючи оригінальну розробку [32] на основі фірмової бібліотеки Xilinx Bitstream Interface та здатність деяких виробів фірми Xilinx здійснювати так звану часткову реконфігурацію (partial reconfiguration), модифікувати зміст таблиць LUT під час функціонування ПЛІС, не заважаючи роботі інших областей інтегральної схеми. Крім того, автори [32] стверджують, що метод дозволяє синтезувати схеми АП на ПЛІС у три-чотири рази ефективніше порівняно з іншими підходами. Але за минулі майже 20 років метод не набув поширення та практичного застосування.

Висновки

Головна перевага ЦК та АП при побудові реконфігуровних засобів інформаційної безпеки — це висока пропускна здатність.

Найсуттєвіший недолік — значне споживання ресурсів, а саме логіки та розподіленої пам'яті (тригерів логічних комірок ПЛІС). Як наслідок — високе енергоспоживання.

Важлива особливість реалізації АП на ПЛІС — занадто високі вимоги щодо здатності навантаження компонентів програмованої логіки та над-

мірна довжина цифрових ліній. Конвеєризація, яка є вирішенням вказаних проблем, призводить до нелінійного зростання апаратних витрат при збільшенні кількості патернів, що обумовлює другий важливий недолік підходу — погану масштабованість за об'ємом словника сигнатур.

Розробники доклали значних зусиль щодо пом'якшення вказаних недоліків. Апаратні витрати підходу дозволяють дещо знизити техніки попереднього дешифрування DCAM та часткового дешифрування DpCAM. Значніше скоротити ресурси дозволяє використання півбайтових компараторів. Оптимізація на бітовому рівні (з використанням апарату БДР) зменшує витрати майже вдвічі, але конфліктує з іншими техніками, зокрема DCAM. Тобто, не зважаючи на докладені зусилля, значних успіхів щодо зниження вартісних показників досягти не вдалося.

Проте досліджений підхід добре масштабується за пропускною здатністю. Розпаралелювання на m потоків призводить до m-кратного прискорення в наслідок майже лінійного зростання витрат, а у поєднанні зі згаданими вище техніками зниження апаратних витрат навіть до масштабування за сублінійним законом.

Проблему поганої масштабованості за об'ємом словника сигнатур вирішують за допомогою кластеризації множини патернів, внаслідок чого велика схема розпізнавання розпадається на кілька підсхем меншого розміру. Застосування спеціального методу кластеризації дозволяє додатково скоротити сукупні апаратні витрати.

Важливим фактом є те, що зміст вхідних даних ніяким чином не впливає на характер роботи схеми розпізнавання на основі АП. Це, по-перше, зумовлює високий рівень такого показника продуктивності, як передбачуваність пропускної здатності, по-друге, робить системи, які використовують даний підхід, невразливими до атак алгоритмічної складності [33], що є одним з функціональних показників. Щодо динамічної реконфігурації, на жаль, такий режим впровадити в схему на ЦК практично неможливо.

Таким чином, отримані результати дозволять розробникам створювати більш ефективні реконфігуровні засоби інформаційної безпеки.

СПИСОК ВИКОРИСТАНОЇ ЛИТЕРАТУРИ

1. *Смит Б.* Методы и алгоритмы вычислений на строках. Теоретические основы регулярных вычислений / Пер. с англ. М.: Вильямс, 2006. — 496 с.
2. *Hilhurt S.Ya.* Application of FPGA-based Reconfigurable Accelerators for Network Security Tasks // Collection of scientific works. Simulation and informational technologies. PIMEE NAS of Ukraine. 2014, Vol. 73, p. 17—26.
3. *Евдокимов В.Ф., Давиденко А.Н., Гильгурт С.Я.* Централизованный синтез реконфигурируемых аппаратных средств информационной безопасности на высокопроизводительных платформах // Захист інформації, 2018, **20**, № 4, с. 247—258.

4. *Гильгурт С.Я.* Применение реконфигурируемых вычислителей для аппаратного ускорения сигнатурных систем защиты информации // Тез. доп. Міжнар. наук.-техн. конф. «Моделювання-2018». Київ: Ін-т проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, 2018, с. 107—110.
5. *Content-Addressable* Memories. 2-nd ed. / T. Kohonen. Berlin, Germany: Springer-Verlag, 1987.
6. *Robinson I.N.* Pattern-addressable memory // IEEE Micro, 1992, Vol. 12, No. 3, p. 20—30.
7. *Pagiamtzis K., Sheikholeslami A.* Content-addressable memory (CAM) circuits and architectures: A tutorial and survey // IEEE Journal of Solid-State Circuits, 2006, Vol. 41, No. 3, p. 712—727.
8. *Neale R.* Is content addressable memory (CAM) the key to network success? // Electronic Engineering, 1999, Vol. 71, No. 865, p. 9—12.
9. *NetLogic* Microsystems [Електронний ресурс]. — Режим доступу: https://web.archive.org/web/20120207195938/http://www.netlogicmicro.com. Загл. з екрану. (Дата звернення: 09.04.2019).
10. *MUSIC-IC* [Електронний ресурс]. Режим доступу: https://www.music-ic.com. Загл. з екрану. (Дата звернення: 09.04.2019).
11. *Guccione S.A., Levi D., Downs D.* A reconfigurable content addressable memory // Parallel and Distributed Processing. Proceedings, 2000, Vol. 1800 LNCS, p. 882—889.
12. *Yu F., Katz R.H., Lakshman T.V.* Gigabit rate packet pattern-matching using TCAM // 12th IEEE International Conference on Network Protocols, Proceedings. 2004, p. 174—183.
13. *Sung J.S., Kang S.M., Lee Y. et al.* A multi-gigabit rate deep packet inspection algorithm using TCAM // IEEE Global Telecommunications Conference (GLOBECOM 05). St Louis, MO: IEEE, 2005, Vol. 1, p. 453—457.
14. *Bispo J., Sourdis L., Cardoso J.M.P., Vassiliadis S.* Regular expression matching for reconfigurable packet inspection // IEEE International Conf. on Field Programmable Technology. Proceedings, (FPT- 2006), Bangkok, Thailand, 2006, p. 119—126.
15. *SNORT* [Електронний ресурс]. Version 2.9.12. Режим доступу: http://www.snort.org. Загл. з екрану. (Дата звернення: 09.04.2019).
16. *ClamAV.* ClamAV® is an open source antivirus engine for detecting trojans, viruses, malware & other malicious threats. [Електронний ресурс]. Version 0.101.2. Режим доступу: http://www.clamav.net. Загл. з екрану. (Дата звернення: 09.04.2019).
17. *Гильгурт С.Я.* Реконфигурируемые вычислители. Аналитический обзор // Электронное моделирование, 2013, 35, № 4, с. 49—72.
18. *Iliopoulos M., Antonakopoulos T.* Reconfigurable network processors based on field programmable system level integrated circuits // 10th International Conf. on Field-Programmable Logic and Applications (FPL-2000). Springer-Verlag, 2000, Vol. 1896, p. 39—47.
19. *A network* instruction detection system on IXP1200 network processors with support for large rule sets / Tech. rep. 2004-02. University Leiden, 2004.
20. *Xinidis K., Anagnostakis K.G., Markatos E.P.* Design and implementation of a high-performance network intrusion prevention system // 20th International Information Security Conference, IFIP/SEC2005. Chiba, 2005, p. 359—374.
21. *Sourdis I., Pnevmatikatos D.N.* Fast, large-scale string match for a 10Gbps FPGA-based network Intrusion Detection System // Field-Programmable Logic and Applications. Proceedings, 2003, Vol. 2778, p. 880—889.
22. *Cho Y.H., Mangione-Smith W.H.* Deep packet filter with dedicated logic and read only memories // 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Proceedings, (FCCM- 2004), Napa, USA, 2004, p. 125—134.

23. *Huang J., Yang Z.K., Du X., Liu W.* FPGA based high speed and low area cost pattern matching // IEEE Region 10 Conference (TENCON 2005). Proceedings. Melbourne, Australia: IEEE, 2005, p. 2693—2697.
24. *Sourdis I., Pnevmatikatos D.N.* Pre-decoded CAMs for efficient and high-speed NIDS pattern matching // 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. Proceedings, 2004, p. 258—267.
25. *Xilinx.* Virtex-II Platform FPGAs: Complete Data Sheet. Product Specification [Електронний ресурс]. Version DS031 (v4.0) April 7, 2014. Режим доступу: https://www.xilinx.com/support/documentation/data_sheets/ds031.pdf. Загл. з екрану. (Дата звернення: 09.04.2019).
26. *Clark C.R., Schimmel D.E.* Efficient reconfigurable logic circuits for matching complex network intrusion detection patterns // Field-Programmable Logic and Applications. Proceedings, 2003, Vol. 2778, p. 956—959.
27. *Clark C.R., Schimmel D.E.* Scalable pattern matching for high speed networks // 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. Proceedings, 2004, p. 249—257.
28. *Sourdis I., Pnevmatikatos D.N., Vassiliadis S.* Scalable multigigabit pattern matching for packet inspection // IEEE Transactions on Very Large Scale Integration Systems (VLSI, 2008), Vol. 16, No. 2, p. 156—166.
29. *Yusuf S., Luk W.* Bitwise optimised CAM for network intrusion detection systems // International Conference on Field Programmable Logic and Applications (FPL-2005), Tampere, 2005, p. 444—449.
30. *Кнут Д.Э.* Искусство программирования. Том 4А. Комбинаторные алгоритмы, часть 1/ Пер. с англ. М.: ООО «И.Д. Вильямс», 2013, 960 с.
31. *Hazelhurst S., Fatti A., Henwood A.* Binary decision diagram representations of firewall and router access lists / Tech. rep. Johannesburg, South Africa: Witwatersrand University, 1998.
32. *Guccione S.A., Levi D.* XBI: A Java-based interface to FPGA hardware // Conference on Configurable Computing — Technology and Applications. Vol. 3526: Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE). Boston, Ma: SPIE-Int Soc Optical Engineering, 1998, p. 97—102.
33. *Гильгурт С.Я., Дурняк Б.В., Коростиль Ю.М.* Противодействие атакам алгоритмической сложности на системы обнаружения вторжений // Моделювання та інформаційні технології// Зб. наук. праць ІПМЕ ім. Г.Є. Пухова НАН України. Київ, 2014, Вип. 71, с. 3—12.

Отримано 10.04.19

REFERENCES

1. Smyth, B. (2003), Computing Patterns in Strings, Pearson Addison Wesley.
2. Hilhurt, S.Ya. (2014), "Application of FPGA-based reconfigurable accelerators for network security tasks", *Simulation and informational technologies PIMEE NAS of Ukraine*, Vol. 73, pp. 17-26.
3. Evdokimov, V.F., Davydenko, A.N. and Hilgurt, S.Ya. (2018), "Synthesis of reconfigurable information security hardware on HPC platforms", *Ukrainian information security research journal*, Vol. 20, no. 4, pp. 247-258.
4. Hilhurt, S.Ya. (2018), "The use of reconfigurable accelerator for speed-up of signature-based information security systems", *Simulation-2018*, Ukraine, Kyiv, PMEE NAS of Ukraine, pp. 107-110.
5. Teuvo, K. (1987), Content-Addressable Memories, Berlin , Germany.
6. Robinson, I.N. (1992), "Pattern-addressable memory", *IEEE Micro*, Vol. 12, no. 3, pp. 20-30.

7. Pagiamtzis, K. and Sheikholeslami, A. (2006), "Content-addressable memory (CAM), circuits and architectures: A tutorial and survey", *IEEE Journal of Solid-State Circuits*, Vol. 41, no. 3, pp. 712-727.
8. Neale, R. (1999), "Is content addressable memory (CAM), the key to network success?", *Electronic Engineering*, Vol. 71, no. 865, pp. 9-12.
9. NetLogic Microsystems, available at: https://web.archive.org/web/20120207195938/, http://www.netlogicmicro.com (accessed May 17, 2019).
10. MUSIC-IC, available at: https://www.music-ic.com (accessed May 17, 2019).
11. Guccione, S.A., Levi, D. and Downs, D. (2000), "A reconfigurable content addressable memory", *Parallel and Distributed Processing, Proceedings*, Vol. 1800, pp. 882-889.
12. Yu, F., Katz, R.H. and Lakshman, T.V. (2004), "Gigabit rate packet pattern-matching using TCAM", *Proceeding of 12th IEEE International Conference on Network Protocols*, 2004, pp. 174-183.
13. Sung, J.S., Kang, S.M. and Lee, Y. (2005), "A multi-gigabit rate deep packet inspection algorithm using TCAM", *Proceeding of IEEE Global Telecommunications Conference (GLOBECOM 05)*, 2005, Vol. 1, pp. 453-457.
14. Bispo, J., Sourdis, L., Cardoso, J.M.P. and Vassiliadis, S. (2006), "Regular expression matching for reconfigurable packet inspectio", *Proceeding of 2006 IEEE International Conference on Field Programmable Technology*, 2006, Bangkok, Thailand, pp. 119-126.
15. SNORT, available at: http://www.snort.org. (accessed May 17, 2019).
16. ClamAV, available at: http://www.clamav.net (accessed May 17, 2019).
17. Hilhurt, S.Ya. (2013), "Reconfigurable accelerators: Analytical review", *Elektronnoye modelirovaniye*, Vol. 35, no. 4, pp. 49-72.
18. Iliopoulos, M. and Antonakopoulos, T. (2000), "Reconfigurable network processors based on field programmable system level integrated circuits C3 - Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)", *Proceeding of 10th International Conference on Field-Programmable Logic and Applications*, (FPL 2000), Vol. 1896, pp. 39-47.
19. Bos, H. and Huang, K. (2004), A network instruction detection system on IXP1200 network processors with support for large rule sets.
20. Xinidis, K., Anagnostakis, K.G. and Markatos, E.P. (2005), "Design and implementation of a high-performance network intrusion prevention system C3 - IFIP Advances in Information and Communication Technology", *Proceeding of 20th International Information Security Conference*, (IFIP/SEC2005), 2005, Chiba, pp. 359-374.
21. Sourdis, I. and Pnevmatikatos, D. (2003), "Fast, large-scale string match for a 10Gbps FPGA-based network Intrusion Detection System", *Proceeding of Field-Programmable Logic and Applications*, 2003, Vol. 2778, pp. 880-889.
22. Cho, Y.H. and Mangione-Smith, W.H. (2004), "Deep packet filter with dedicated logic and read only memories", *Proceeding of 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004, Napa, USA, pp. 125-134.
23. Huang, J., Yang, Z.K., Du, X. and Liu, W. (2005), "FPGA based high speed and low area cost pattern matching", *Proceeding of IEEE Region 10 Conference* (TENCON 2005), 2005, Nov 21-24, Melbourne, Australia, pp. 2693-2697.
24. Sourdis, I. and Pnevmatikatos, D. (2004), "Pre-decoded CAMs for efficient and high-speed NIDS pattern matching", *Proceeding of 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004, pp. 258-267.
25. Xilinx, "Virtex-II Platform FPGAs: Complete Data Sheet. Product Specification", available at: https://www.xilinx.com/support/documentation/data_sheets/ds031.pdf (accessed May 17, 2019).

26. Clark, C.R. and Schimmel, D.E. (2003), "Efficient reconfigurable logic circuits for matching complex network intrusion detection patterns", *Proceeding of Field-Programmable Logic and Applications*, 2003, Vol. 2778, pp. 956-959.
27. Clark, C.R. and Schimmel, D.E. (2004), "Scalable pattern matching for high speed networks", *Proceeding of 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004, pp. 249-257.
28. Sourdis, I., Pnevmatikatos, D.N. and Vassiliadis, S. (2008), "Scalable multigigabit pattern matching for packet inspection", *IEEE Transactions on Very Large Scale Integration* (VLSI), Systems, Vol. 16, no. 2, pp. 156-166.
29. Yusuf, S. and Luk, W. (2005), "Bitwise optimised CAM for network intrusion detection systems", *Proceedings of International Conference on Field Programmable Logic and Applications*, 2005, Tampere, pp. 444-449.
30. Knut, D.E. (2011), The Art of Computer Programmin, Vol. 4A, Combinatorial Algorithms, part 1, Vilyams, Moscow, Russia.
31. Hazelhurst, S., Fatti, A. and Henwood, A. (1998), Binary decision diagram representations of firewall and router access lists, Johannesburg , South Africa.
32. Guccione, S.A. and Levi, D. (1998), "XBI: A Java-based interface to FPGA hardware", *Proceedings of the Society of Photo-Optical Instrumentation Engineers* (SPIE), 1998, Boston, Soc Optical Engineering, Vol. 3526, pp. 97-102.
33. Hilgurt, S.Ya., Durnyak, B.V. and Korostil, Yu.M. (2014), "Intrusion detection systems defense against algorithmic complexity attacks", *Modelyuvannya ta informatsiyni tekhnolohiyi*, Vol. 71, pp. 3-12.

Received 10.04.19

С.Я. Гильгурт

ПОСТРОЕНИЕ АССОЦИАТИВНОЙ ПАМЯТИ НА ЦИФРОВЫХ КОМПАРАТОРАХ РЕКОНФИГУРИРУЕМЫМИ СРЕДСТВАМИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Сетевые системы обнаружения вторжений, работа которых основана на сигнатурном анализе, выполняют в реальном времени ресурсоемкую задачу одновременного поиска множества заданных строк символов в интенсивном потоке данных. Традиционные программные решения уже не удовлетворяют современным требованиям к их быстродействию. Поэтому все более популярными становятся аппаратные ускорители на основе ПЛИС. Один из наиболее распространенных подходов построения быстродействующих распознающих схем на программируемой логике основан на применении ассоциативной памяти и цифровых компараторов, из которых она состоит. Для повышения эффективности создаваемых реконфигурируемых средств информационной безопасности проанализированы преимущества и недостатки такого подхода, особенности его реализации на ПЛИС, возникающие проблемы и пути их решения.

Ключевые слова: защита информации, сигнатурный анализ, ПЛИС, ассоциативная память, цифровой компаратор, эффективность.

S.Ya. Hilgurt

CONSTRUCTING CAMS ON DIGITAL COMPARATORS BY RECONFIGURABLE MEANS FOR SOLVING INFORMATION SECURITY TASKS

Such information security means as Network Intrusion Detection Systems (NIDS) inspect the network packet payload to search malicious content. This process, deep packet inspection, involves detection of predefined signature strings. A computationally intensive task of string matching becomes a bottleneck of network defense facilities. Since conventional software-based string matching tools have not kept pace with the increasing network speeds, hardware solutions bases on Field Programmable Gate Arrays (FPGAs) have been introduced to solve this problem. There are several different methods for constructing hardware matching schemes on FPGAs. One of the most popular methods is based on Content-Addressable Memory (CAM) and underlying digital comparators. In this paper, a comprehensive analysis of this method is fulfilled. The key features of CAMs, their pros and cons, specifics of realization in hardware as well as encountered problems and ways to overcome them are investigated in details. The results obtained contribute to the effective constructing FPGA-based information security means.

K e y w o r d s: NIDS, DPI, string matching, FPGA, CAM, digital comparator.

ГІЛЬГУРТ Сергій Якович, канд. техн. наук, ст. наук. співр. Ін-ту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України. В 1986 р. закінчив Київський інститут інженерів цивільної авіації (м. Київ). Область наукових досліджень — розпізнавальні системи, реконфігуровні обчислення на основі програмованої логіки, технічні засоби інформаційної безпеки локальних обчислювальних мереж.