УДК 658.512.011:681.326:519.713

# ИНФРАСТРУКТУРА АНАЛИЗА И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КИБЕРПРОСТРАНСТВА

**Хаханов В.И., Литвинова Е.И.**
Харьковский национальный университет радиоэлектроники
кафедра автоматизации проектирования вычислительной техники
E-mail: hahanov@kture.kharkov.ua

***Аннотация***

***Хаханов В.И., Литвинова Е.И. Инфраструктура анализа и информационной безопасности киберпространства.*** *Предложена модель и метрика кибернетического пространства, где субъектами выступают взаимодействующие процессы или явления с физическим носителем в виде компьютерных систем и сетей. Разработана оценка меры бинарного отношения двух объектов в киберпространстве для распознавания любого типа взаимодействия объектов при решении практических задач информационной безопасности. Предложена модель быстродействующего гетерогенного мультиматричного процессора, предназначенного для быстрого поиска информационных объектов в киберпространстве. Описаны инфраструктуры сервисного обслуживания программного продукта для защиты от вредоносных программ и верификации проектов цифровых систем. Предложена структурная модель распознавания деструктивных образов на основе отношений входного образа, эталонов и критериев сходства.*

## Введение

Тенденция последних лет в части создания новых коммуникационных, вычислительных и информационных сервисов, полезных для человека, обращает внимание на создание все более специализированных гаджетов (gadget), обладающих существенными преимуществами перед персональными компьютерами и ноутбуками: энергопотребление, компактность, вес, стоимость, функциональные возможности, дружественность интерфейса. Практически вся десятка лучших за 2010 год специализированных изделий (Apple iPad, Samsung Galaxy S, Apple MacBook Air, Logitech Revue, Google Nexus One (HTC Desire), Apple iPhone 4, Apple TV, Toshiba Libretto W100, Microsoft Kinect, Nook Color) реализована в виде цифровых систем на кристаллах. К 2012 году рынок мобильной и беспроводной связи и перейдет на 20 нм (итоги январского Саммита 2011 альянса Common Platform). Дальнейшее развитие технологий по годам: 2014 – 14 нм, 2016 – 11 нм. В 2015 году 55% сотовых телефонов станут смартфонами, планшетные компьютеры заменят ноутбуки и нетбуки. Суперфоны (Nexus-1, Google) станут той соединительной тканью, которая свяжет все остальные устройства и сервисы. Надвигается следующая волна компьютеризации под названием «интернет вещей», которая приведет к широкому распространению датчиковых сетей, включая их интеграцию в человеческое тело. Мировой рынок перечисленных выше устройств и гаджетов насчитывает сегодня порядка 3 миллиардов изделий.

С учетом изложенного выше можно сделать следующие выводы в отношении эволюционирования киберпространства: 1) Персональный компьютер, уходя с рынка, трансформируется в широкий спектр гаджетов доступа в киберпространство, обладающих функциональностью персональных компьютеров, компактностью и низкой стоимостью. 2) Как интерфейс связи между человеком и киберпространством гаджет в меньшей степени нуждается в защите. 3) Киберпространство имеет иерархию от индивидуального дружественного пространства пользователя до глобального, где фигурируют «облака», данные и сети по интересам. 4) Для повышения информационной безопасности предметного

(целевого) киберпространства возможно делать его блуждающим, что затруднит определение его точного адреса. 5) Нарождается новая структура в виде киберпространства как части общей экосистемы планеты, для которой необходимо создавать инфраструктуру информационной безопасности как средство сервисного обслуживания. 6) Стремительно развивается мощный сегмент рынка планшетных компьютеров, которые не имеют входов и выходов, кроме Internet. Такая ситуация подталкивает пользователя к созданию индивидуального киберпространства, не зависимого от типа гаджета, которое должно быть надежно защищено. 7) Сегодня еще высокая стоимость программных приложений на компьютере экономически оправдывает применение антивирусных защит. Но завтра, для дешевых устройств типа, IPad и IPhon, с низкой мощностью программных приложений антивирусная защита станет экономически нецелесообразной. Одновременно возникает новый субъект для защиты – индивидуальное киберпространство (данные и приложения), которое получает сервисы от технологических «облаков» и сетей по интересам. 8) Одним из возможных решений для охраны пространства может быть его вакцинация в виде введения некоторой избыточности, которая дает возможность осуществлять мониторинг и экстренную связь с облаком антивирусных сервисов в целях удаления функциональных нарушений. 9) Вакцинация данных и программ на персональном компьютере также может быть интересным решением. Она позволяет более оперативно осуществлять мониторинг данных, используя информацию от внедренных в папки и программы агентов в виде ассерционных операторов.

Цель публикации – существенное повышение качества сервисов, доставляемых со стороны программных, аппаратных изделий, сетевых структур киберпространства и уменьшение стоимости эксплуатационных расходов за счет создания инфраструктур сервисного обслуживания и безопасности, обеспечивающих дружественную эксплуатацию, тестирование и устранение нефункциональных деструктивных компонентов. Для достижения цели необходимо решить такие задачи: 1) Разработка модели кибернетического пространства. 2) Математический аппарат и двигатель для анализа и сервисного обслуживания киберпространства. 3) Процесс-модели и критерии взаимодействия вредоносных компонентов с программными кодами полезных функциональностей. 4) Инфраструктура киберпространства и реализация ее компонентов.

Источники: актуальные проблемы анализа киберпространства [1]; метрика киберпространства [2]; аппаратура и матричные процессоры для поиска информации [3]; распознавание деструктивных образов при защите киберпространства [4].

**Метрика киберпространства**

Пусть имеется $n \neq 0$ конечное число точек в пространстве, составляющее цикл, где каждая из них задана двоичным вектором, длины k:

$$A = (A_1, A_2, ..., A_i, ..., A_n) = \{(a_{11}, a_{12}, ..., a_{1j}, ..., a_{1k}), (a_{21}, a_{22}, ..., a_{2j}, ..., a_{2k}), ...,$$
$$, ..., (a_{i1}, a_{i2}, ..., a_{ij}, ..., a_{ik}), ..., (a_{n1}, a_{n2}, ..., a_{nj}, ..., a_{nk})\}, a_{ij} = \{0,1\}.$$

Расстояние между двумя точками определяется в виде:

$$d_i = d_i(A_i, A_{i+1}) = a_{i,j} \overset{k}{\underset{j=1}{\oplus}} a_{i+1,j}.$$

Метрика $\beta$ кибернетического или векторного логического двоичного пространства определяется нулевой xor-суммой расстояний $d_i$ между ненулевым и конечным числом точек, замкнутых в цикл:

$$\beta = \overset{n}{\underset{i=1}{\oplus}} d_i = 0.$$

**Двигатель для киберпространства**

Для скоростного путешествия по киберпространству (поиск объектов и оценка их взаимодействия) – нужен простой и быстродействующий мультиматричный процессор (ММП), где каждая команда (and, or, xor, slc) обрабатывает параллельно и предельно быстро только одну бинарную операцию на матрицах (двумерные массивы данных). Количество командно-ориентированных матриц-примитивов создает систему – гетерогенный мультиматричный процессор бинарных операций с буфером М, рис. 1. Мультиматричный модуль процессора включает 4 блока памяти со встроенными на них операциями (A – and, B – xor, C – slc – shift left crowding, D – or) и буферную память М. Модуль ориентирован на параллельное выполнение в данном случае одной из четырех инструкций (ISA – Instruction Set Architecture), оперирующих матрицами двоичных данных одинаковой размерности: $M = M\{and, or, xor, slc\}\{A, B, C, D\}$ с занесением результата в буфер М.

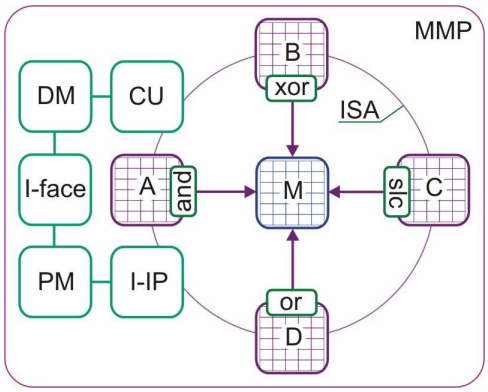


Рис. 1. Мультиматричный процессор бинарных операций

Особенность ММП в том, что не ячейка матрицы имеет систему команд из четырех операций, а каждая команда имеет собственную матрицу ячеек в качестве данных для параллельной обработки, что существенно упрощает структуру управления и устройства в целом. Вся сложность ММП перенесена на структуры данных, где память матрицы имеет одну аппаратно-реализованную встроенную команду, что позволяет иметь примитивную систему управления параллельными вычислительными процессами (SIMD – Single Instruction Multiple Data), последовательностную по своей сути, а значит, нет необходимости создавать сверхсложные компиляторы, ориентированные на распараллеливание вычислительных процессов. Здесь каждый матричный процессор выполняет одну операцию, встроенную в запоминающие элементы матрицы. Но возникают ситуации, когда матричный уровень (M-level) задания данных избыточен для выполнения, например операций над булевыми (B-level) или регистровыми (R-level) переменными. Для такого случая необходимо иметь иерархию по уровням представления данных. Стандартные блоки ММП: памяти данных DM и программ PM, управления CU, интерфейс I-face и сервисного обслуживания I-IP, а также мультиматричный модуль процессора, включающий 4 блока памяти со встроенными в них операциями (A – and, B – xor, C – or, D – slc – shift left crowding) и буферную память М.

**Модель поиска вредоносных компонентов в программных продуктах**

Аналитическая модель верификации HDL-кода с использованием механизма темпоральных ассерций (дополнительных линий наблюдения) ориентирована на достижение заданной глубины диагностирования и представлена в следующем виде:

$$\begin{aligned} &M = f(F, A, B, S, T, L), && F = (A * B) \times S;\ S = f(T, B);\\ &A = \{A_1, A_2, ..., A_i, ..., A_n\}; && B = \{B_1, B_2, ..., B_i, ..., B_n\};\\ &S = \{S_1, S_2, ..., S_i, ..., S_m\}; && S_i = \{S_{i1}, S_{i2}, ..., S_{ij}, ..., S_{ip}\};\\ &T = \{T_1, T_2, ..., T_i, ..., T_k\}; && L = \{L_1, L_2, ..., L_i, ..., L_n\}. \end{aligned} \quad (3)$$

Здесь $F=(A*B)\times S$ – функциональность, представленная графом (рис. 2) транзакций программных блоков (Code-Flow Transaction Graph – CFTG), где $S=\{S_1,S_2,...,S_i,...,S_m\}$ – вершины или состояния программного продукта при моделировании тестовых сегментов. Каждое состояние $S_i=\{S_{i1},S_{i2},...,S_{ij},...,S_{ip}\}$ определяется значениями существенных переменных проекта (булевы, регистровые переменные, память). Ориентированные дуги графа представлены конкатенацией программных блоков $B=(B_1,B_2,...,B_i,...,B_n)$ и соответствующих им ассерций $A=\{A_1,A_2,...,A_i,...,A_n\}$. Каждая дуга $B_i$ – группа операторов кода – формирует состояние вершины $S_i=f(T,B_i)$ в зависимости от теста $T=\{T_1,T_2,...,T_i,...,T_k\}$. Вершина может иметь более одной входящей (исходящей) дуги. Множество блоков с функциональными нарушениями представлено списком $L=\{L_1,L_2,...,L_i,...,L_n\}$.

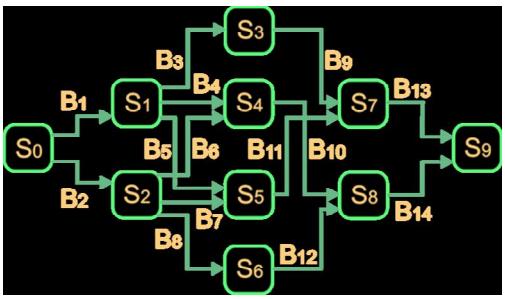


Рис. 2. Пример ABC-графа программы

Модель кода, представленная в форме Assertion Based Coverage Graph (ABC-Graph) отображает не только структуру программного кода, но и тестовые срезы функциональных покрытий, формируемые с помощью программных блоков, входящих в рассматриваемую вершину.

**Инфраструктура тестирования вредоносных компонентов и спама**

На основе мультиматричного (-регистрового) процессора создана инфраструктура (рис. 3) тестирования кода программных систем, которая является модификацией I-IP стандарта 1500 [2]. Здесь фигурируют четыре процесс-модели: тестирование на стадии моделирования, диагностирование функциональных нарушений, оптимизация диагноза, восстановление работоспособности. Процесс-модель тестирования включает модель кода, механизм ассерций, testbench и coverage (покрытие). Последнее оценивает качество теста проверки всех состояний проекта. В результате моделирования синтезируется матрица активизации программных блоков B и матрица ассерционных реакций A на тестовые сегменты, которая может быть трансформирована к вектору состояния ассерций m путем применения функции or к вектор-столбцам A-матрицы:

$$\begin{cases} B=(T\oplus F); \\ m=\bigvee\limits_{j=1}^{m} A_j \leftarrow A=(T\oplus A^c). \end{cases}$$

Два последних компонента используются во второй процесс-модели для диагностирования блоков кода. Результатом диагностирования является вектор дефектов, формирующий подмножество блоков $m_d$ с функциональными нарушениями. При этом не исключены ошибки как в testbench, так и в ассерционных операторах, отвечающих за тестирование и мониторинг программных блоков. Тройственная неопределенность диагноза $D=\{B_j,T_i,A_{ij}\}$ характерна при отсутствии точной идентификации блока в процедуре сравнения столбцов матрицы активизации с вектором ассерционных реакций. Третий блок решает задачу минимизации числа блоков, подозреваемых в наличии функциональных

нарушений, до одного из них. При этом используется матрица активизации блоков и диагноз $m_d$, полученный в предыдущей процесс модели. Модуль исправления функциональных нарушений ориентирован на ручной поиск ошибок в одном программном блоке, представленном вектором $m_b$. Возможен также автоматический режим исправления ошибок в блоках, если в инфраструктуре верификации предусмотрена библиотека диверсных программных модулей, имеющих аналогичные функциональности. Предложенная инфраструктура является одним из шагов на пути создания автомата тестирования программных блоков.

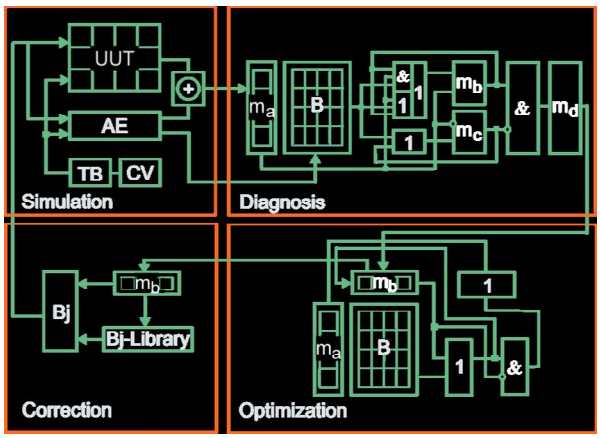


Рис. 3. Инфраструктура верификации HDL-кода

**Выводы**

1. Предложена модель эволюционирующего кибернетического пространства, где субъектами выступают взаимодействующие процессы или явления с физическим носителем в виде компьютерных систем и сетей. Стандартизация пространства и всех взаимодействующих субъектов, включая негативные, возможна на основе предложенной бета метрики, которая структурировано и адекватно оценивает меру взаимодействия отношений в киберпространстве.

2. Разработаны инфраструктуры сервисного обслуживания программного продукта для защиты от вредоносных программ и верификации проектов цифровых систем, которая включает шесть компонентов, создающих условия для защиты и надежного функционирования программных и аппаратных компонентов.

3. Разработана архитектура мультиматричного процессора, ориентированного на повышение быстродействия процедур встроенного диагностирования функциональных нарушений в программном или аппаратном изделии, которая отличается использованием параллельных логических векторных операций and, or, xor, slc операций, что дает возможность существенно (x10) повысить быстродействие диагностирования одиночных и/или кратных дефектов (функциональных нарушений).

4. Предложена инфраструктура диагностирования вредоносных компонентов программного кода цифровых, которая имеет четыре процесс-модели для тестирования, диагностирования, оптимизации и исправления ВК, замкнутые в цикл, что дает возможность уменьшить время сервисного обслуживания.

**Литература**

1. Babulak E. Future Global Office // 12th International Conference “Computer Modelling and Simulation”.– 2010.– P. 352–356. 2. Инфраструктура мозгоподобных вычислительных процессов / М.Ф. Бондаренко, О.А. Гузь, В.И. Хаханов, Ю.П. Шабанов-Кушнаренко.– Харьков: Новое Слово.– 2010.– 160 с. 3. Shibata T. Implementing brain-like systems using nano functional devices // Ultimate Integration of Silicon, ULIS 2009.– 2009.– P. 131-134. 4. Hilewitz

Y., Lauradoux C., Lee R.B. Bit matrix multiplication in commodity processors // International Conference Application-Specific Systems, Architectures and Processors.– 2008.– P. 7-12.