УДК 004.02

А. Б. Сизоненко, В. В. Меньших

ИСПОЛЬЗОВАНИЕ СЕТЕЙ ПЕТРИ ДЛЯ МОДЕЛИРОВАНИЯ СПОСОБОВ РАСПАРАЛЛЕЛИВАНИЯ АЛГОРИТМОВ ЗАЩИТЫ ИНФОРМАЦИИ В СИСТЕМАХ С МАССИВНО-ПАРАЛЛЕЛЬНЫМИ СОПРОЦЕССОРАМИ

Цель исследований – разработка формализованного способа реализации алгоритмов защиты информации в многопроцессорных вычислительных системах и оценка их эффективности. Анализ архитектур современных вычислительных систем показал, что в большинстве вычислительных систем, кроме центрального, имеются сопроцессоры, предназначенные для выполнения специфических задач. Это могут быть графические или арифметические сопроцессоры, ресурсы которых не всегда используются полностью. Предлагается неиспользуемые ресурсы сопроцессоров использовать для повышения производительности программных средств защиты информации, функционирование которых связано с выполнением логических вычислений. Для этого разработан алгоритм, предусматривающий анализ архитектуры вычислительной системы, реализацию логических вычислений системой команд вычислителя, выделение независимых ветвей в алгоритме, определение трудоемкости вычислений каждого функционального блока и алгоритма в целом. Даже если система команд не поддерживает логические вычисления, имеется возможность их реализации на таком вычислителе. Это возможно при использовании математического аппарата представления логических функций арифметическими полиномами. Производительность программных средств защиты информации предложено определять путем моделирования с использованием временных сетей Петри. Рассмотрен пример построения сети Петри для моделирования реализации алгоритма шифрования ГОСТ 28147-89 в системе с центральным процессором и арифметическим сопроцессором. Определена трудоемкость вычислений функций отдельных блоков алгоритма ГОСТ 28147-89 при их реализации арифметическими полиномами. Эти значения являются исходными данными для построения модели и определяют время непримитивных событий. В качестве среды моделирования используется система CPN Tools. Результаты моделирования показали, что производительность повышается на 4 % при использовании ресурсов одноядерного арифметического сопроцессора.

Ключевые слова: булевы функции, системы булевых функций, параллельные логические вычисления, алгоритмы, сети Петри, CPN Tools.

Введение

В настоящее время увеличиваются объемы информации, обрабатываемой в информационных и телекоммуникационных системах. Это предоставление государственных услуг в электронном виде, развитие системы ситуационных центров, повсеместное введение электронного документооборота, использование сети видеоконференцсвязи. В то же время увеличивается и вероятность рисков, связанных с существованием угроз безопасности информации. В связи с этим возникает необходимость разработки высокопроизводительных программных, аппаратных и аппаратно-программных средств защиты информации от различных типов угроз. Достаточно большое количество средств и систем защиты информации используют в своей работе интенсивные логические вычисления. Это криптографические средства защиты информации, средства защиты от ошибок, системы разграничения доступа, а также процессы моделирования средств и систем защиты информации.

Современные средства вычислительной техники часто строятся по разнесенной архитектуре. Большинство ЭВМ имеют в своем составе кроме центрального процессора еще и вспомогательные вычислители, предназначенные для выполнения специфических задач. Например, в каждой ЭВМ имеется графический сопроцессор, серверы могут содержать арифметические сопроцессоры. Таким образом, возникает задача максимально использовать ресурсы всех вычислительных устройств для выполнения функций защиты информации.

Нами предлагается последовательность действий по определению оптимального способа представления и вычисления логических функций средств и алгоритмов защиты информации. Оценку загрузки вычислителей при выполнении реализации средств и алгоритмов защиты информации предлагается осуществлять с использованием сетей Петри.

Описание многопроцессорных систем с массивно-параллельными сопроцессорами

Сопроцессор представляет собой специализированный процессор, расширяющий и дополняющий возможности центрального процессора. Это может быть отдельная микросхема (модуль), либо сопроцессор может быть встроен в центральный процессор. Сопроцессор расширяет систему команд центрального процессора и делает выполнение некоторых инструкций более производительным. Сопроцессоры эффективны при решении задач, обладающих параллелизмом по данным, число арифметических операций в которых велико по сравнению с операциями ветвления и обращения к памяти [1].

Математические сопроцессоры находят достаточно широкое применение и используются для ускорения операций над числами с плавающей точкой (Floating Point Unit, FPU). В первых x86-совместимых процессорах это была отдельная микросхема (Intel 8087/80287/80387), в более поздних (начиная с Intel 80486) блок FPU встроен в процессор.

Известен отечественный арифметический сопроцессор К1810ВМ87, работающий совместно с центральным процессором К1810ВМ86 [2]. Он рассчитан на работу в системах с интенсивной численной обработкой, в которых численные данные изменяются в очень широком диапазоне, возникают очень большие и очень малые промежуточные результаты, требуется высокая точность вычислений, необходима производительность, превышающая возможности центрального процессора.

Еще один арифметический сопроцессор, Л1839ВМ2Ф, входит в состав микропроцессорного комплекта Л1839 и функционирует совместно с микропроцессором Л1839ВМ1Ф. Он предназначен для выполнения команд умножения и деления целых чисел и всех команд обработки чисел с плавающей запятой [3]. С учетом уровня развития современной микроэлектроники сопроцессор имеет очень скромные характеристики (тактовая частота – 10 МГц; время выполнения микрокоманды – 200 нс). Несмотря на это, микропроцессорный комплект выпускается в настоящее время [4] и находит свое применение в военной, космической промышленности, на их основе строятся специализированные ЭВМ для транспорта, авиации, рассчитанные на тяжелые условия эксплуатации, например «Элинс-36» [5], а также авиационные бортовые вычислительные машины БЦВМ-90 [6]. Это продиктовано необходимостью использования в таких системах отечественной элементной базы для исключения недекларированных возможностей аппаратной части.

Можно привести в качестве примера и современный арифметический сопроцессор ClearSpeed X700 [7]. Архитектура CSX предназначена для выполнения высокопроизводительных операций над числами с плавающей точкой. В отличие от центрального процессора, CSX 700 имеет небольшие массогабаритные показатели и низкое тепловыделение (всего 9 Вт) при производительности 96 GFLOPS с числами с одинарной и двойной точностью, работает на невысокой тактовой частоте (250 МГц).

Каждый процессор (рис. 1) содержит два независимых многопоточных SIMD-массивных процессора (Multi-threaded SIMD array processors, MTAP). В свою очередь, каждый MTAP содержит устройство управления (Control Unit, CU), моноисполнительный блок (Mono Execution Unit, MEU) и полиисполнительный блок (Poly Execution Unit, PEU). Архитектура MTAP обеспечивает мощные и масштабируемые вычисления, основанные на SIMD-массиве из процессорных элементов (Processing Elements, PEs).

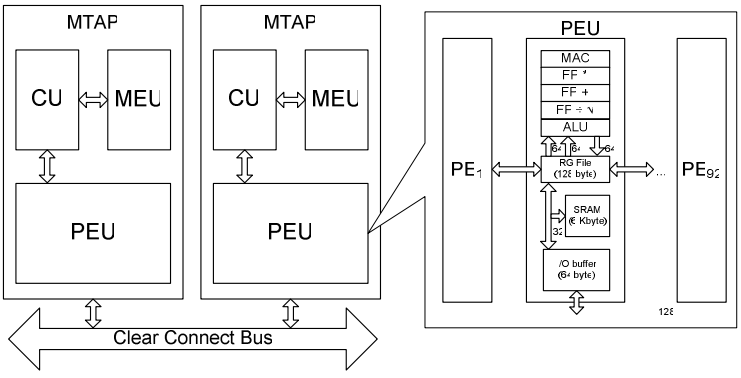

Рис. 1. Структура арифметического сопроцессора CSX 700

MEU отвечает за обработку скалярных и непараллельных данных, ветвлений и переключение потоков. В то же время на аппаратном уровне поддерживает многопоточность (до 8 потоков). MEU содержит арифметико-логическое устройство (ALU), 64-разрядный блок вычислений с плавающей точкой (FPU), несколько 128-байтных регистров.

PEU содержит массив из 96 PEs, обеспечивающих синхронное выполнение множественного набора данных (SIMD). Каждый PEs включает в себя несколько блоков обработки и имеет высокий уровень внутреннего параллелизма на уровне команд и данных, имеет свою собственную локальную память (128 байтный регистр, 6 Кбайт RAM), содержит блок вычисления с плавающей точкой с одинарной и двойной точностью, с конвейерной обработкой сложения и умножения, блок поддержки целочисленных вычислений, деления и вычисления квадратного корня.

Графический процессор (Graphic Processing Unit, GPU) является также массивно-параллельным сопроцессором центральному процессору (CPU). Последовательный код выполняется на CPU, а для массивно-параллельных вычислений используется GPU как набор одновременно выполняющихся нитей (threads). CPU состоит из нескольких ядер, предназначенных для последовательной обработки данных, в то время как GPU состоит из тысяч ядер, предназначенных для параллельной обработки данных. Ядра CPU выполняют один поток последовательных инструкций с максимальной производительностью (MIMD), а GPU проектируются для быстрого исполнения большого числа параллельно выполняемых потоков инструкций. Универсальные процессоры оптимизированы для достижения высокой производительности единственного потока команд, обрабатывающего и целые числа, и числа с плавающей точкой. Для эффективной загрузки GPU необходимы тысячи потоков (нитей), в то время как для CPU это значение составляет 10–20 [8]. Вычислительная модель графического сопроцессора представлена на рис. 2.

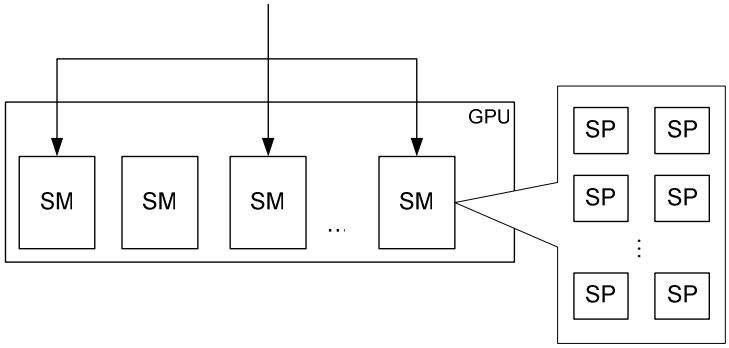

Рис. 2. Структура графического сопроцессора

Графический процессор представляет собой набор независимых потоковых мультипроцессоров (Streaming Multiprocessor, SM), каждый из которых состоит из нескольких скалярных процессоров или ядер (Scalar Processor, SP), предназначенных для выполнения операций с числами с плавающей точкой. Кроме скалярных процессоров, потоковый мультипроцессор может содержать блоки вычисления специальных функций (Special Function Unit, SFU), блок управления командами (IU) и собственную память. В потоковых мультипроцессорах последних поколений содержится блок для обработки 64-битных данных с плавающей точкой (Double Precision Unit) [1, 8, 9].

Современный графический процессор основан на SIMT-архитектуре (Single Instruction, Multiple Thread). На аппаратном уровне потоки разбиваются на свертки (warps) по 32 потока. Внутри сверток все потоки выполняют одни и те же инструкции. Если в пределах свертки осуществляется ветвление, то все потоки свертки выполняют все возможные пути. Это негативно сказывается на производительности и при программировании на графических процессорах необходимо стремиться, чтобы в пределах свертки потоки выполняли одинаковые инструкции [9]. Потоки разных сверток (warps) могут находиться на разных стадиях выполнения [8]. Все запущенные потоки организованы в иерархию: сетка – блок – поток (рис. 3).

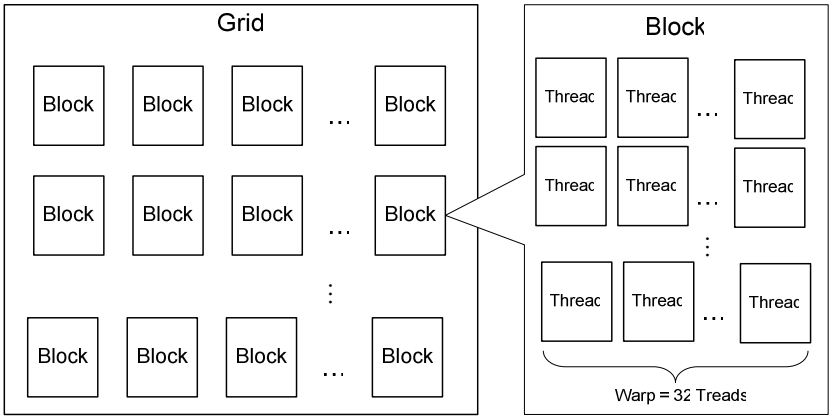

Рис. 3. Иерархия потоков

Блоки потоков (Blocks) объединяются в решетки блоков (Grids). Решетка представляет собой одномерный или двухмерный массив блоков, каждый блок – одно-, двух- или трехмерный массив нитей [8]. Потоки из разных блоков не могут эффективно взаимодействовать между собой.

Можно выделить следующие особенности выполнения вычислений на массивно-параллельных сопроцессорах:

– могут одновременно выполнять большое количество потоков инструкций;

– эффективно выполняют арифметические операции над числами с плавающей точкой;

– эффективно выполняют однотипные инструкции, ветвление негативно сказывается на производительности.

Алгоритм повышения эффективности использования массивно-параллельных сопроцессоров для решения задач защиты информации

Шаг 1. Анализ архитектуры вычислительной системы, определяются количество и вид вычислителей, входящих в ее состав. В результате анализа необходимо получить следующие сведения, которые станут исходными данными для следующих шагов:

1. Система команд, поддерживаемая вычислителем.

2. Количество ядер в вычислителе.

3. Разрядность.

Шаг 2. Представление логических функций средств и систем защиты информации в базисе, поддерживаемом системой команд вычислителя.

При табличном способе задания булевой функции (табл. 1) каждому набору аргумента приписывается определенное значение функции [10]. Наборы и соответствующие им значения группируются в таблицу. Достоинством такого способа представления является наибольшая скорость вычисления значений булевой функции или системы булевых функций, недостатком – то, что для хранения всех значений необходим большой объем памяти.

Булева функция может быть представлена в виде алгебраического выражения суперпозиции элементарных логических операций – в виде формул [11].

В нормальных формах последовательно выполняются не более двух базовых операций [12]. В совершенных нормальных формах (СНФ), кроме того, все члены имеют одинаковую размерность [12].

Анализ источников [10–16] показывает, что в общем случае совершенные полиномиальные формы можно описать формулой

$$f(X) = \overset{2^n-1}{\underset{i=0}{*}} a_i x_{n-1}^{i_{n-1}} \circ \ldots \circ x_1^{i_1} \circ x_0^{i_0},$$

где i_{n-1}, ..., i_1, i_0 – разряды при двоичном представлении числа i. Название формы, операции * и ○, степенная операция x_j^i, область значений коэффициента a_i представлены в табл. 1.

Таблица 1

Полиномиальные формы представления булевых функций

Форма	Операция *	Операция ○	Степенная операция	a_i
Конъюнктивная	$\wedge$	$\vee$	$x_j^i = \begin{cases} x_j, i=0 \\ \overline{x}_j, i=1 \end{cases}$	$a_i \in \{0,1\}$
Дизъюнктивная	$\vee$	$\wedge$	$x_j^i = \begin{cases} \overline{x}_j, i=0 \\ x_j, i=1 \end{cases}$	$a_i \in \{0,1\}$
Жегалкина	$\oplus$	$\wedge$	$x_j^i = \begin{cases} 1, i=0 \\ x_j, i=1 \end{cases}$	$a_i \in \{0,1\}$
Арифметическая	+	*	$x_j^i = \begin{cases} 1, i=0 \\ x_j, i=1 \end{cases}$	$a_i \in Z$

В минимизированной нормальной форме (МНФ) количество первичных термов минимально и последовательно выполняется не более двух базовых операций алгебры логики. Для получения МНФ из СНФ применяется аналитический метод минимизации или графический (диаграммы Вейча) [12].

При вынесении в нормальных формах общих членов за скобки количество последовательно выполняемых операций, необходимых для вычисления значения функции (порядок функции), увеличивается. Такие формы называют скобочными [12].

Если несколько функций вычисляются на одном наборе аргументов, то для более эффективного использования разрядности процессора используются векторные вычисления, являющиеся разновидностью SIMD. Функцию в этом случае целесообразно представить в обобщенной форме [13] и выполнять действия с целочисленными коэффициентами $A_i \in Z$.

Пусть дана система булевых функций:

$$\begin{cases} f_0(X) = \overset{2^n-1}{\underset{i=0}{*}} a_{0i} x_{n-1}^{i_{n-1}} \circ \ldots \circ x_1^{i_1} \circ x_0^{i_0}; \\ f_1(X) = \overset{2^n-1}{\underset{i=0}{*}} a_{1i} x_{n-1}^{i_{n-1}} \circ \ldots \circ x_1^{i_1} \circ x_0^{i_0}; \\ \vdots \\ f_{d-1}(X) = \overset{2^n-1}{\underset{i=0}{*}} a_{(d-1)i} x_{n-1}^{i_{n-1}} \circ \ldots \circ x_1^{i_1} \circ x_0^{i_0}. \end{cases}$$

Для получения обобщенной формы необходимо для вычисления каждой функции выделить свой разряд. Для этого происходит сдвиг функции на определенной количество разрядов влево, что соответствует умножению каждого коэффициента a_{ji} функции $f_j(X)$, на 2^j $(j = 1, \ldots, d-1)$. Далее выполняется приведение подобных слагаемых и получение обобщенного полинома:

$$F(X) = \overset{2^n-1}{\underset{i=0}{*}} A_i x_{n-1}^{i_{n-1}} \circ \ldots \circ x_1^{i_1} \circ x_0^{i_0}.$$

Вычисление обобщенного полинома производится поразрядным суммированием коэффициентов A_i, если операция * логическая, и по правилам арифметического сложения, если $* \Rightarrow +$.

Так как массивно-параллельные сопроцессоры, входящие в состав ПЭВМ, в основном ориентированы на выполнение арифметических операций, целесообразно использовать арифметические способы задания булевых функций.

Шаг 3. Выделение независимых ветвей в алгоритме защиты информации, которые могут быть реализованы с учетом использования системы команд процессора и сопроцессора.

Шаг 4. Определение скорости вычисления независимых ветвей алгоритма защиты информации, реализованных на шаге 2 в системе команд вычислителя.

Шаг 5. Моделирование функционирования вычислительной системы с учетом задействования для выполнения части операций алгоритма защиты информации массивно-параллельного вычислителя. Результаты моделирования должны дать ответ на вопрос, эффективным ли будет задействование сопроцессора для решения задач защиты информации и насколько.

Применение сетей Петри для моделирования параллельных логических вычислений в системах с арифметическим сопроцессором

Для моделирования процесса распределенных вычислений удобно использовать математический аппарат сетей Петри [17], а именно класс иерархических временных раскрашенных сетей Петри.

На примере алгоритма ГОСТ 28147-89 рассмотрим возможность его реализации в системе с распределенной архитектурой (центральный процессор и арифметический сопроцессор) в системе с микропроцессорным комплектом Л1839.

Результаты представления сетями Петри процесса зашифрования сообщения в многопроцессорной системе в общем виде приведены на рис. 4.

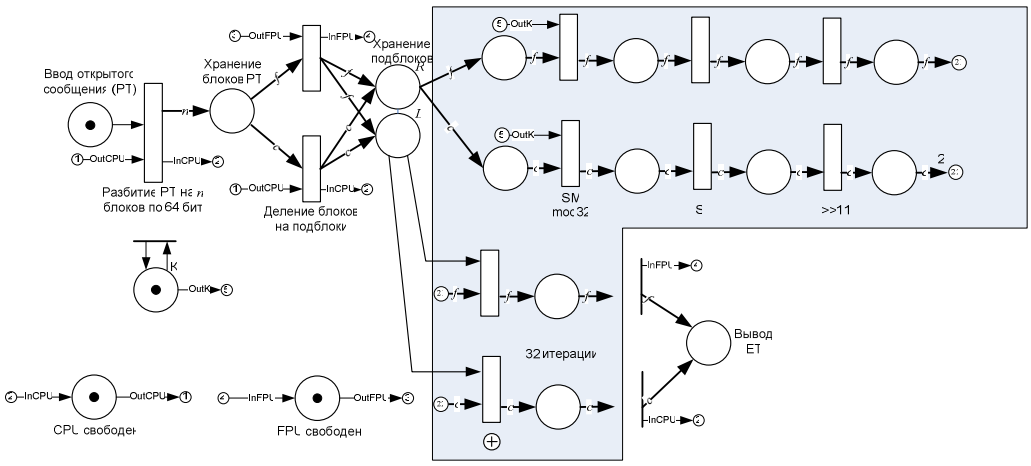

Рис. 4. Сеть Петри, моделирующая процесс зашифрования сообщения блочным шифром ГОСТ 28147-89 в многопроцессорной системе (символом ▯ обозначается непримитивное событие): CPU – центральный процессор; FPU – арифметический сопроцессор; InCPU, OutCPU, InFPU, OutFPU – соответственно входы и выходы состояния, обозначающего, что центральный процессора и сопроцессор находятся в режиме ожидания; PT (plaintext) открытое сообщение; OutK – цикловые подключи; SM mod32 – сумматор по модулю 32; >>11 – циклический сдвиг вправо на 11 бит; S – блок подстановок; c, f – кратность ребер для центрального процессора и сопроцессора соответственно

Для исключения излишней громоздкости при отображении сети Петри введены обозначения, например ①. Это означает, что фишка из позиции «CPU свободен» может пойти по одной из дуг, отмеченных также символом ①.

Кратность ребер c и f определяется количеством ядер в центральном процессоре и сопроцессоре соответственно.

Необходимо определить длительность непримитивного события. Так как сопроцессор имеет арифметический набор команд, то и функции указанных блоков должны быть реализованы с использованием арифметического набора команд. Алгоритм ГОСТ 28147-89 содержит такие типовые блоки, как подстановки, суммы по модулю 32, циклического сдвига и суммы по модулю два [10].

Порядок представления логических функций посредством арифметических полиномов приведен в [13, 14, 16], способы реализации подстановок и перестановок показаны в [13, 18, 19].

Блоки нелинейной замены (S-блоки), имеющие n входов и d выходов ($n \geq d$) могут быть представлены арифметическим полиномом длины 2^n.

Функция сдвига вправо представима линейным арифметическим полиномом n переменных с разрядностью коэффициентов d [13].

Сумма по модулю два заменяется операцией арифметического сложения. Однако при векторной реализации операций арифметического сложения понадобится в 2 раза большая разрядность, т. к. необходимо предусмотреть дополнительный разряд для переноса.

С учетом вышеизложенного количество операций, необходимых для обработки информации алгоритмом ГОСТ 28147-89, при осуществлении вычислений центральным процессором и сопроцессором, приведено в табл. 2.

Таблица 2

Количество операций для одного раунда ГОСТ 28147-89

Блок	Количество операций для CPU	Количество операций для FPU
SM mod32	1	1
8 S-блоков 4 × 4	8	8 × 32 = 256
>>11	1	31
L $\oplus$ R (32 бит)	1	2
Итого для одного раунда	11	9280

Таким образом, для шифрования одного блока (32 раунда) понадобится 352 и 9280 операций соответственно.

В университете Орхуса (Aarhus Universitet, Дания) была разработана и свободно распространяется для некоммерческого использования система моделирования CPN Tools. Для исследования моделей раскрашенных временных сетей Петри разработан интуитивно понятный графический интерфейс, предусмотрен специальный язык на основе языка CPN ML для описания запросов [20]. Система CPN Tools позволяет создавать, редактировать модели, анализировать поведение моделей с помощью имитации динамики сети Петри.

На рис. 5 показаны результаты моделирования распределенной обработки информации алгоритмом ГОСТ 28147-89 в системе с одноядерными центральным процессором и арифметическим сопроцессором.

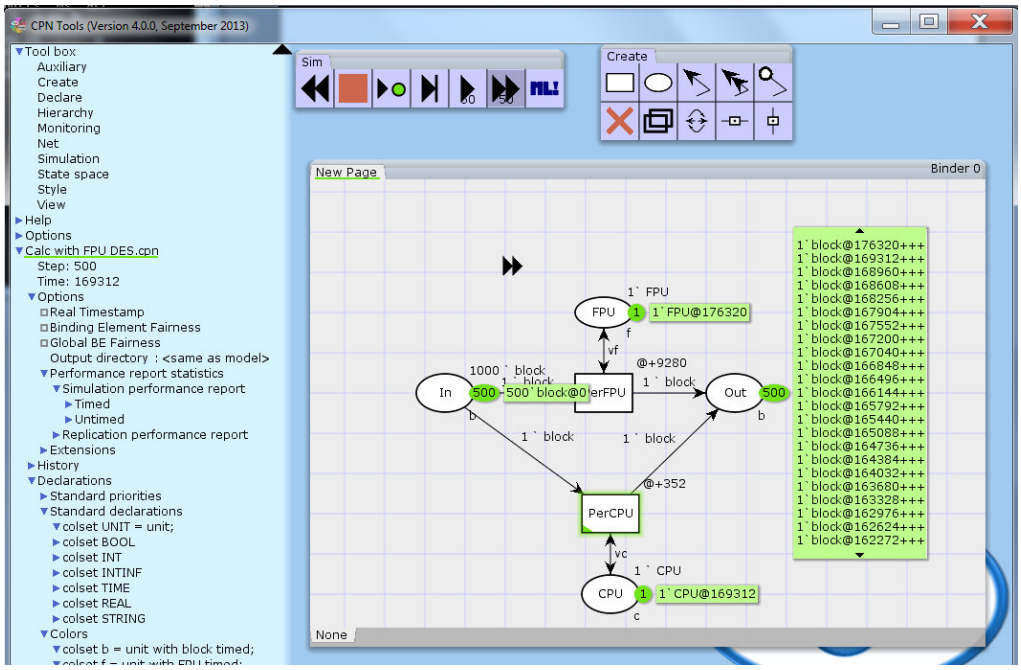

Рис. 5. Результат моделирования распределенной обработки информации алгоритмом ГОСТ 28147-89 в вычислительной системе с арифметическим сопроцессором

В представленной модели используются фишки трех «цветов»: b – для обозначения блоков исходного текста; c – для обозначения центрального процессора; f – для обозначения арифметического сопроцессора.

Определены 4 позиции:

In – вход, на который поступает открытое сообщение, разбитое на блоки, для зашифрования. Для примера в модели взята 1000 блоков (фишек цвета b).

FPU и CPU – арифметический сопроцессор и центральный процессор соответственно свободен. Так как в рассматриваемом примере процессоры одноядерные, то таких позиций также по одной.

Out – выход модели, куда поступают блоки после обработки.

В модели два перехода – PerCPU и PerFPU, обозначающие обработку информации процессором и сопроцессором соответственно. Для моделирования времени обработки блоков центральным процессором и сопроцессором в переходах используются временные метки после префикса @. В рассматриваемом примере введено вычисленное количество операций, необходимых для обработки одного блока центральным процессором и арифметическим сопроцессором.

Запустив построенную модель на выполнение, получаем, что для обработки 500 блоков открытого текста с использованием центрального процессора и сопроцессора понадобится 169 312 временных интервалов (тактов функционирования). При обработке только центральным процессором этого же количества блоков понадобится 176 000 временных интервалов.

Заключение

Моделирование процессов обработки информации алгоритмами защиты информации в многопроцессорных системах позволит избежать использования дорогостоящего оборудования для оценки эффективности этих вычислений.

На простейшем примере нами показана возможность применения системы CPN Tools для моделирования процессов распределенной обработки информации блочными алгоритмами шифрования. Результаты моделирования более сложных систем (например, многопроцессорной многоядерной) позволят разработчику более эффективно использовать вычислительные ресурсы ЭВМ. Для этого в системе моделирования CPN Tools имеется возможность строить сложные иерархические сети с учетом взаимосвязей между ними.

Кроме того, при моделировании реальных вычислительных систем должны учитываться статистические данные по загрузке вычислительных устройств для выполнения задач по прямому назначению и имеющих больший приоритет.

СПИСОК ЛИТЕРАТУРЫ

1. *Боресков А. А.* Параллельные вычисления на GPU. Архитектура и программная модель CUDA: учеб. пособие / А. В. Боресков, А. А. Харламов, Н. Д. Марковский. М.: Изд-во Моск. ун-та, 2012. 336 с.

2. *Григорьев В. Л.* Архитектура и программирование арифметического сопроцессора / В. Л. Григорьев. М.: Энергоатомиздат, 1991. 208 с.

3. *Микросхема* интегральная Л1839ВМ2: техническое описание. 179 с.

4. *НПО Ангстрем* // URL: http://www.angstrem.ru/.

5. *АО «Научно-технический центр* «ЭЛИНС» // URL: http://www.aha.ru.

6. *ОАО «Уфимское* приборостроительное производственное объединение» // URL: http://www.uppo.ru/.

7. *URL: http*://www.clearspeed.com/.

8. *Боресков А. В.* Основы работы с технологией CUDA / А. В. Боресков, А. А. Харламов. М.: ДМК Пресс, 2013. 232 с.

9. *Линев А. В.* Технологии параллельного программирования для процессоров новых архитектур: учеб. / А. В. Линев, Д. К. Боголепов, С. И. Бастраков; под ред. В. П. Гергеля. М.: Изд-во Моск. ун-та, 2010. 160 с.

10. *Фомичев В. М.* Дискретная математика и криптология: Курс лекций / В. М. Фомичев; под общ. ред. Н. Д. Подуфалова. М.: Диалог-МИФИ, 2003. 400 с.

11. *Закревский А. Д.* Логические основы проектирования дискретных устройств / А. Д. Закревский, Ю. В. Поттосин, Л. Д. Черемисинова. М.: ФИЗМАТЛИТ, 2007. 592 с.

12. *Пухальский Г. И.* Цифровые устройства: учеб. пособие для втузов / Г. И. Пухальский, Т. Я. Новосельцева. СПб.: Политехника, 1996. 885 с.

13. *Малюгин В. Д.* Параллельные логические вычисления посредством арифметических полиномов / В. Д. Малюгин. М.: Наука. Физматлит, 1997. 192 с.

14. *Шалыто А. А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов / А. А. Шалыто. СПб.: Наука, 2000. 747 с.

15. *Выхованец В. С.* Обработка сигналов в дискретных базисах на основе обобщенных полиномиальных форм / В. С. Выхованец // Докл. 2-й Междунар. конф. «Цифровая обработка сигналов и ее применение». М., 1999. Т. 2. С. 372–377.

16. *Финько О. А.* Модулярная арифметика параллельных логических вычислений: моногр. / О. А. Финько; под ред. В. Д. Малюгина. М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2003. 224 с.

17. *Питерсон Дж.* Теория сетей Петри и моделирование систем / Дж. Питерсон. М.: Мир, 1984. 264 с.

18. *Сизоненко А. Б.* Параллельная реализация криптографических блоков подстановок и перестановок арифметическими полиномами / А. Б. Сизоненко // Докл. Том. гос. ун-та систем управления и радиоэлектроники. 2012. № 2 (26), ч. 1. С. 140–144.

19. *Сухарева Е. М.* Криптографические методы защиты информации / Е. М. Сухарева, Е. М. Сухарев, В. М. Амербаев, Р. Г. Бияшев, Ю. В. Виланский; под ред. Е. М. Сухарева. М.: Радиотехника, 2007. Кн. 4. 312 с.

20. *Зайцев Д. А.* Моделирование телекоммуникационных систем в CPN Tools / Д. А. Зайцев, Т. Р. Шмелева. Одесса: ОНАС им. А. С. Попова, 2009. 72 с.

Статья поступила в редакцию 27.05.2014

ИНФОРМАЦИЯ ОБ АВТОРАХ

Сизоненко Александр Борисович – Россия, 350005, Краснодар; Краснодарский университет Министерства внутренних дел Российской Федерации; канд. техн. наук, доцент; начальник кафедры «Информационная безопасность»; siz_al@mail.ru.

Меньших Валерий Владимирович – Россия, 394065, Воронеж; Воронежский институт Министерства внутренних дел Российской Федерации; д-р физ.-мат. наук, профессор; начальник кафедры «Высшая математика»; menshikh@list.ru.

A. B. Sizonenko, V. V. Menshikh

USE OF PETRI NETS FOR MODELING THE METHODS OF PARALLELIZATION OF INFORMATION SECURITY ALGORITHMS IN THE SYSTEMS WITH MASSIVE PARALLEL COPROCESSORS

Abstract. The purpose of the work is to develop the formalized way of realization of the algorithms of information security in the multiprocessor computing systems and an assessment of their efficiency. The analysis of the architecture of the modern computing systems showed that in the majority of the computing systems, except central processing unit, there are the coprocessors intended for the performance of the specific tasks. It can be the graphic or arithmetic coprocessors, the resources of which are not always used completely. It is offered to use the unused resources of the coprocessors to increase the productivity of software of the information security, functioning of which is associated with the performance of the logical calculations. For this purpose, the algorithm providing the analysis of the architecture of the computing system, realization of the logical calculations by the system of commands of the calculator, allocation of the independent branches in the algorithm, determination of the labor input of the calculations of each functional block and the algorithm as a whole, is worked out. Even if the system of commands does not support the logical calculations, there is a possibility of their realization in such a calculator. It is possible while using the mathematical apparatus of representation of the logical functions by arithmetic polynomials. It is offered to determine the performance of software for information protection by simulation using timed Petri nets. An example of creation of Petri network for modeling of realization of the enciphering algorithm of State Standard 28147-89 in the system with the central processor and the arithmetic coprocessor is reviewed. The complexity of the computing functions of the individual blocks of the algorithm of State Standard 28147-89 is defined at their realization by arithmetic polynomials. These values are the initial data for the creation of the model and define time of imprimitive events. As the modeling environment the CPN Tools system is used. The results of modeling showed that the productivity increases by 4 percent when using the resources of the single-core arithmetic coprocessor.

Key words: Boolean functions, system of Boolean functions, parallel logical computing, algorithms, Petri nets, CPN Tools.

REFERENCES

1. Boreskov A. V., Kharlamov A. A., Markovskii N. D. *Parallel'nye vychisleniia na GPU. Arkhitektura i programmnaia model' CUDA* [Parallel computing at GPU. Architecture and program model CUDA]. Moscow, Izdatel'stvo Moskovskogo universiteta, 2012. 336 p.
2. Grigor'ev V. L. *Arkhitektura i programmirovanie arifmeticheskogo soprotsessora* [Architecture and programming of arithmetic coprocessor]. Moscow, Energoatomizdat, 1991. 208 p.
3. *Mikroskhema integral'naia L1839VM2: tekhnicheskoe opisanie* [Integral microchart L1839BM2: technical characteristics]. 179 p.
4. *NPO Angstrem* [SIA Angstrem]. Available at: http://www.angstrem.ru/.
5. *AO «Nauchno-tekhnicheskii tsentr ELINS»* [JS "Scientific engineering center ELINS"]. Available at: http://www.aha.ru.
6. *OAO «Ufimskoe priborostroitel'noe proizvodstvennoe ob"edinenie»* [JSC "Ufimsk machine building industrial organization"]. Available at: http://www.uppo.ru/.
7. *Available at*: http://www.clearspeed.com/.
8. Boreskov A. V., Kharlamov A. A. *Osnovy raboty s tekhnologiei CUDA* [Bases of the operational technology CUDA]. Moscow, DMK Press, 2013. 232 p.
9. Linev A. V., Bogolepov D. K., Bastrakov S. I. *Tekhnologii parallel'nogo programmirovaniia dlia protsessorov novykh arkhitektur* [Technologies of parallel programming for the processors of new architectures]. Pod redaktsiei V. P. Gergelia. Moscow, Izd-vo Moskovskogo universiteta, 2010. 160 p.
10. Fomichev V. M. *Diskretnaia matematika i kriptologiia: Kurs lektsii* [Discrete mathematics and cryptology: Theoretical course]. Pod obshchei redaktsiei N. D. Podufalova. Moscow, Dialog-MIFI, 2003. 400 p.
11. Zakrevskii A. D., Pottosin Iu. V., Cheremisinova L. D. *Logicheskie osnovy proektirovaniia diskretnykh ustroistv* [Logical bases of designing the discrete tools]. Moscow, FIZMATLIT, 2007. 592 p.
12. Pukhal'skii G. I., Novosel'tseva T. Ia. *Tsifrovye ustroistva* [Digital devices]. Saint-Petersburg, Politekhnika, 1996. 885 p.
13. Maliugin V. D. *Parallel'nye logicheskie vychisleniia posredstvom arifmeticheskikh polinomov* [Parallel logical computing by means of arithmetic polynomials]. Moscow, Nauka. Fizmatlit Publ, 1997. 192 p.
14. Shalyto A. A. *Logicheskoe upravlenie. Metody apparatnoi i programmnoi realizatsii algoritmov* [Logical control. Methods of apparatus and program realization of the algorithms]. Saint-Petersburg, Nauka Publ., 2000. 747 p.
15. Vykhovanets V. S. Obrabotka signalov v diskretnykh bazisakh na osnove obobshchennykh polinomial'nykh form [Processing of the signals in discrete basis of general polynomial forms]. *Doklady 2-i Mezhdunarodnoi konferentsii «Tsifrovaia obrabotka signalov i ee primenenie»*. Moscow, 1999, vol. 2, pp. 372–377.
16. Fin'ko O. A. *Moduliarnaia arifmetika parallel'nykh logicheskikh vychislenii* [Modular arithmetic of parallel logical computing]. Pod redaktsiei V. D. Maliugina. Moscow, Institut problem upravleniia im. V. A. Trapeznikova Rossiiskoi akademii nauk, 2003. 224 p.
17. Piterson Dzh. *Teoriia setei Petri i modelirovanie sistem* [Theory of Petri nets and system modeling]. Moscow, Mir Publ., 1984. 264 p.
18. Sizonenko A. B. Parallel'naia realizatsiia kriptograficheskikh blokov podstanovok i perestanovok arifmeticheskimi polinomami [Parallel realization of cryptographic blocks of substitutes and replacements of arithmetic polynomials]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniia i radioelektroniki*, 2012, no. 2 (26), part 1, pp. 140–144.
19. Sukhareva E. M., Sukharev E. M., Amerbaev V. M., Biiashev R. G., Vilanskii Iu. V. *Kriptograficheskie metody zashchity informatsii* [Cryptographic methods of information protection]. Pod redaktsiei E. M. Sukhareva. M.: Radiotekhnika Publ., 2007. Kn. 4. 312 p.
20. Zaitsev D. A., Shmeleva T. R. *Modelirovanie telekommunikatsionnykh sistem v CPN Tools* [Modeling of tele-communication systems in CPN Tools]. Odessa, ONAS imeni A. S. Popova, 2009. 72 p.

The article submitted to the editors 27.05.2014

INFORMATION ABOUT THE AUTHORS

Sizonenko Alexander Borisovich – Russia, 350005, Krasnodar; Krasnodar University of the Ministry of Internal Affairs of the Russian Federation; Candidate of Technical Sciences, Assistant Professor; Head of the Department "Information Security"; siz_al@mail.ru.

Menshikh Valeriy Vladimirovich – Russia, 394065, Voronezh; Voronezh Institute of the Ministry of Internal Affairs of the Russian Federation; Doctor of Physics and Mathematics, Professor; Head of the Department "Mathematics"; menshikh@list.ru.