УДК 681.3

Мультиконвейерная архитектура высокопроизводительных криптоблоков, используемых в составе «систем на кристалле»

И.И. Шагурин, Г.Ю. Жихарев

Национальный исследовательский ядерный университет МИФИ, IIShagurin@mephi.ru

***Аннотация* — Для обеспечения достоверности и конфиденциальности информации, представляемой в электронной форме, используется ее криптографическое шифрование. Широкое применение имеют криптоалгоритмы, выполняющие вычисление хеш-суммы, которая используется для идентификации сообщений, выявления изменения их содержания, быстрого поиска информации и банках данных, формирования цифровой подписи. Для реализации этих алгоритмов предлагается мультиконвейерная архитектура криптоблоков, которая значительно повышает их производительность. Описывается микроархитектура исполнительного конвейера, приводятся результаты моделирования криптоблока. Дается сравнительная оценка эффективности предлагаемых архитектурных решений по критерию «производительность/требуемые ресурсы».**

***Ключевые слова* — криптоалгоритм, криптоблок, хеш – сумма, исполнительный конвейер, мультиконвейерная архитектура, пропускная способность.**

I. Введение

Наиболее распространенными алгоритмами вычисления хеш-суммы являются MD-5, SHA-1 и SHA-2/256, в соответствии с которыми формируется хеш-сумма (хеш) сообщения в виде двоичного кода длиной 128, 180 или 256 бит. Этот код является уникальным для каждого сообщения и определяется путем разбиения сообщения на блоки размером 512 бит (16 слов M0,… M15 по 32 бита) и выполнения последовательности операций их обработки [1], [2]. Обработка блока производится в течение одного или четырех раундов, в каждом из которых выполняется *n* последовательных итераций получения хеша. На каждой итерации в соответствии с заданным алгоритмом выполняется последовательность операций сложения, сдвига и логического преобразования исходных 32-битных хеш-операндов (*A, B, … G, H*) и 32-битных слов M_i, входящих в состав обрабатываемого блока. При реализации алгоритма MD5 используются операнды *A, B, C, D*, при реализации SHA-1 - операнды *A, B, C, D, E*, при реализации SHA-2/256 - операнды *A, B, C, D, E, F, G, H*. Для получения хеша требуется 64 или 80 итераций.

Криптоблоки, реализующие алгоритмы хеширования, используются в ряде микроконтроллеров, а также входят в состав «систем на кристалле» (СнК), реализуемых в виде СБИС или на базе программируемых логических микросхем типа FPGA.

Для приложений, требующих быстрой обработки большого потока сообщений, важное значение имеет производительность (пропускная способность) криптоблока. Эффективным методом повышения производительности является организация конвейерного выполнения итераций вычисления хеша. В работах [3] - [7] описаны различные способы организации конвейера, выполняющего эти итерации. Однако для реализации алгоритмов хеширования требуется также производить преобразования слов обрабатываемого блока данных на каждой *i* – й итерации, начиная с *i* = 16. В таблице 1 указаны: *L* - число логических операций И, ИЛИ, НЕ, Исключающее ИЛИ; *S* – число операций суммирования; *R* – число циклических и логических сдвигов операндов, которые выполняются на одной итерации для вычисления текущих значений хеша (*H*) и необходимого преобразования слов данных (*D*). Таким образом, синхронно с последовательностью операций вычисления хеша требуется выполнять последовательность операций преобразования слов сообщения.

Таблица 1

Сравнение сложности алгоритмов

Алгоритм	Число раундов	Число **n** итераций	Разрядность хеша, бит	L(H)	S(H)	R(H)	L(D)	S(D)	R(D)
MD-5	4	16	4 x 32	2 – 5 *		1	Перестановка слов**		
SHA-1	4	20	5 x 32	2 – 5 *		2	3		6
SHA-2/256	1	64	8 x 32	13		6	4		6
* Количество логических операций для алгоритмов MD5, SHA-1 зависит от номера раунда. ** На каждой итерации, начиная с i = 16, изменяется последовательность выборки слова данных.									

В статье [8] описана мультиконвейерная архитектура криптоблока, содержащего набор кольцевых исполнительных конвейеров (ИК), каждый из которых содержит синхронно работающие параллельные линии – конвейер вычисления хеша (КВХ) и конвейер подготовки данных (КПД). При полной загрузке конвейеров криптоблок может обрабатывать непрерывный поток сообщений, формируя на выходе хеши поступающих блоков данных в каждом рабочем такте. В данной работе предлагается модифицированный вариант мультиконвейерной архитектуры, в котором сбалансирована вычислительная нагрузка ступеней ИК, что позволяет повысить тактовую частоту и, соответственно, пропускную способность криптоблока.

II. Мультиконвейерная архитектура криптоблоков

Наиболее сложные арифметико-логические преобразования выполняются при вычислении хеша по алгоритму SHA-2/256. При этом на каждой итерации вычисляются новые значения хеш-операндов A и E в соответствии с функциями:

$$F = L1(A,B,C) + R1(A) + (Mi + Ki),$$

$$T(A) = H + L0(E,F,G) + R0(E) + F, \quad (1)$$

$$P(E) = D + H + F, \quad (2)$$

где *L0, L1* – логические функции трех переменных, *R0*, *R1* – логические функции над циклически сдвинутыми операндами, *Mi* – слово блока данных, *Ki* – константа, выбираемая в соответствии с алгоритмом и номером итерации. Для остальных операндов меняется только их размещение: $A \to B$, $B \to C$, $C \to D$, $E \to F$, $F \to G$, $G \to H$.

Выбор операндов и выполняемых функций определяется заданным алгоритмом. При реализации алгоритма SHA-1 на каждой итерации вычисляются новые значения операнда *A*:

$$T(A) = E + L(B,C,D) + (A \text{ rotl } 5) + (Mi + Ki). \quad (3)$$

Операнд B передается с циклическим сдвигом влево, B *rotl30* $\to C$ остальные операнды передаются без изменения: $A \to B$, $C \to D$, $D \to E$.

При реализации алгоритма MD5 вычисляются новые значения операнда *B*:

$$T(B) = B + ((A + L(B,C,D) + (Mi + Ki)) \text{ rotl } N), \quad (4)$$

где N - число разрядов циклического сдвига влево, которое зависит от номера раунда и итерации. Остальные операнды передаются без изменения: $B \to C$, $C \to D$, $D \to A$. Вид логической функции *L(B,C,D)* зависит от выбора алгоритма и номера раунда.

Данные, приведенные в таблице 1, показывают, что основная вычислительная нагрузка ложится на КВХ, длительность рабочего такта которого будет определять максимальную частоту синхронизации конвейеров и производительность криптоблока. Функция *T*, определяющая сложность комбинационной схемы вычисления хеша, для всех трех алгоритмов содержит операцию суммирования *(Mi + Ki)*, слагаемые которой доступны до начала вычисления. Данную операцию можно выполнить до начала очередной итерации вычисления хеша. В структуре ИК реализован сдвиг стадий КВХ и КПД и выполнение операции *(Mi + Ki) = MKi* на предыдущей стадии КПД (рис. 1). При этом в КВХ уменьшается число последовательно выполняемых операций суммирования (табл. 1), с которыми связаны основные затраты времени.

Предлагаемая мультиконвейерная архитектура криптоблока показана на рис. 1. В каждом рабочем такте на вход КВХ поступает очередное слово *MKi*. После полной загрузки ИК в нем одновременно на разных ступенях обрабатываются шестнадцать 512-битных блоков данных. Получение хеша для одного блока требует 4 или 5 циклов прохождения ИК.

Рис. 1. Мультиконвейерная архитектура криптоблока

Синхронизация КВХ и КПД производится в соответствии с цикличностью их функционирования. Преобразование слов для всех трех криптоалгоритмов выполняется с периодом 16 итераций и начинается после первых шестнадцати итераций. Таким образом, псевдораунд для подготовки данных *MKi* в КПД содержит 16 итераций. Обработка блока данных с помощью алгоритмов MD5, SHA-2/256 потребует четырех псевдораундов, с помощью SHA-1 пяти псевдораундов. Раунды вычисления хеша для MD5 меняются с периодичностью 16 итераций, для SHA-1 – 20 итераций.

В составе криптоблока содержится системный контроллер (СК), выполняющий формирование и распределение блоков данных *MWi* между ИК и задающий режим их работы в соответствии с заданным алгоритмом, выбор которого определяется кодом ALG. При получении кода каждая стадия ИК конфигурируется для выполнения функций в соответствии с номером текущего раунда.

Для корректной обработки блоков данных СК формирует заголовок каждого блока LB, который содержит (рис. 2): NR = 0, 1, 2 или 3 – номер выполняемого раунда (для SHA-2/256 псевдораунда), SX – признак выполнения алгоритма SHA-1 (SX = 1), Z – признак выполнения начального псевдораунда (Z = 0), L – признак последнего блока в сообщении (L = 1), V – признак наличия в блоке загруженных данных (V=1), ID – идентификатор обрабатываемого блока и сообщения.

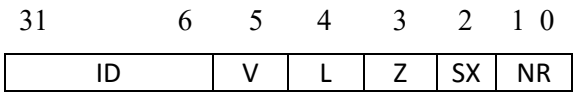

Рис. 2. Формат заголовка блока данных

В исходном состоянии для всех блоков *MWi* в КПД устанавливается значение V = 0. Для блоков, загружаемых из СК в КПД, устанавливаются NR=0, Z = 0, V = 1. Признак SX устанавливается в соответствии с выполняемым алгоритмом, значения L, ID определяются расположением блоков в обрабатываемых сообщениях. СК контролирует значения признаков в LB блока, поступающего с выхода последней стадии КПД. При поступлении блока, имеющего V = 1, SX = 0, NR < 3, СК увеличивает на 1 значение NR и запускает блок в КПД для выполнения следующего раунда. При значении NR = 3 в конвейер загружается новый блок данных. Если блок, поступивший с выхода КПД, имеет признаки V = 1, NR < 3 и SX = 1 (алгоритм SHA-1), то СК пропускает его на вход КПД, не меняя значение NR. В этом случае необходимое изменение NR и SX выполняют схемы модификации MNR, входящие в состав стадий SD3, SD7, SD11 конвейера КПД. При этом для блока, имеющего NR = 3, на стадии SD11 устанавливается признак SX = 0 и при поступлении этого блока на вход СК он заменяется новым блоком. Таким образом, при реализации SHA-1 обеспечивается задержка выполнения каждого раунда в КВХ на 4 итерации после окончания предыдущего псевдораунда КПД. Для загрузки в СК блоков данных используется высокоскоростной интерфейсный модуль. В состав СК входит буферная память для промежуточного хранения блоков.

После загрузки в один из ИК 16 блоков данных СК последовательно направляет следующие 16 блоков в другой конвейер. После прохождения четырех конвейерных циклов (раундов) обработка одного поступившего блока по алгоритму MD5 или SHA-2/256 завершается и вместо него в ИК вводится новый блок данных. При непрерывном поступлении блоков в четырех ИК одновременно обрабатываются 64 блока данных. Полная загрузка всех четырех ИК при выполнении алгоритмов MD5, SHA-2/256 обеспечивает на выходе криптоблока формирование хеша для одного 512-битного блока данных в каждом такте работы конвейера. Алгоритм SHA-1, требующий большего числа псевдораундов, реализуется с пониженной на 25% скоростью вычисления хеша.

Выходной мультиплексор выбирает с выходов КВХ полученные значения хешей для блоков данных, которые имеют значения V = 1, NR = 3 (обработка закончена). При этом СК определяет выбор исполнительного конвейера в соответствии с порядком их загрузки. Блок формирования хеша содержит сумматор-аккумулятор, который суммирует поступающие значения хешей для блоков одного сообщения. Накопление хеш-суммы сообщения заканчивается после обработки последнего блока, имеющего признак L = 1. Промежуточное хранение хеш-сумм осуществляется в буферной памяти. Полученные хеш-суммы выдаются вместе с идентификатором сообщения.

III. МИКРОАРХИТЕКТУРА ИСПОЛНИТЕЛЬНОГО КОНВЕЙЕРА

При проектировании микроархитектуры КПД и КВХ решалась задача равномерного распределения вычислительной нагрузки между ступенями конвейеров для обеспечения максимальной частоты их функционирования. Так как наибольшие затраты времени связаны с выполнением суммирования, анализировались возможности сокращения количества последовательно включенных сумматоров с распространением переноса (CPA) и их замены на более простые и быстрые сумматоры с сохранением переноса (CSA). Проведена оценка характеристик регистров и комбинационных устройств, выполняющих преобразование 32-разрядных операндов. При оценке использованы данные, полученные при логическом синтезе на базе библиотеки КМОП-элементов с проектными нормами 65 нм. Результаты оценки приведены в таблице 2, где значения потребляемой мощности даны для частоты переключения 1 ГГц, задержка переключения указана в относительных единицах, Тз – средняя задержка переключения логических вентилей И-НЕ, ИЛИ-НЕ, Исключающее ИЛИ.

Таблица 2

Характеристики основных элементов схемы

	Площадь, мкм2	Потребляемая мощность, мкВт	Задержка переключения
Регистр	890	570	(2 – 3) Тз
Сумматор CPA, 2 операнда	1510	870	(5 – 6) Тз
Сумматор CSA, 3 операнда	580	510	(2 – 3) Тз
Логическое устройство: программируемая функция 3 переменных фиксированная функция 3 переменных	 1290 590	 550 410	 (2 – 3) Тз (2 – 3) Тз
Мультиплексор «2 в 1» Мультиплексор «4 в 1»	430 880	190 310	(2 – 3) Тз (2 – 3) Тз
Программируемый сдвигатель на N разрядов* (4 варианта выбора N)	3520	1240	(2 – 3) Тз
*) Сдвиг на любое фиксированное число разрядов реализуется путем соответствующего соединения разрядных линий и не требует дополнительных затрат времени и ресурсов			

При средних значениях Тз =50 пс для обеспечения частоты синхронизации 1 ГГц с запасом 20% длина критических цепей в каждой ступени конвейера должна быть не более 800/50 = 16 вентилей. Число последовательно включенных сумматоров CPA в ступенях КВХ, КПД не должно превышать двух. Для реализации функции T на каждой стадии КВХ требуется суммирование пяти операндов, поэтому для выполнения итерации требуются 2 ступени конвейера. С учетом этого проведено распределение по ступеням КВХ устройств, реализующих функции вычисления хеша согласно алгоритму SHA-2/256. Полученная микроархитектура стадии SHi (рис. 3) содержит регистры A, B, …, G, H для хранения хеш-операндов, сумматоры CPA, CSA, логические блоки, выполняющие операции с тремя операндами - *A, B, C* или *E, F, G*:

L0 = (A and B) xor (A and C) xor (B and C),
L1 = (E and F) xor ((not E) and G),
и с циклически сдвинутыми вправо значениями одного операнда - *A* или *E*:

R0 = (A rotr 2) xor (A rotr13) xor (A rotr 22),
R1 = (E rotr 6) xor (E rotr 11) xor (E rotr 25).

Верхняя вычислительная ветвь R0/L0-CPA2-CSA2-CPA5 реализует функцию *T(A)*, нижняя R1/L1/CPA1-CSA1-CPA3-CPA4 – функцию *P(E)*. Коммутаторы – мультиплексоры K1, K2 обеспечивают подключение входов и выходов регистров хранения операндов (со сдвигом или без него) в соответствии с выполняемым алгоритмом.

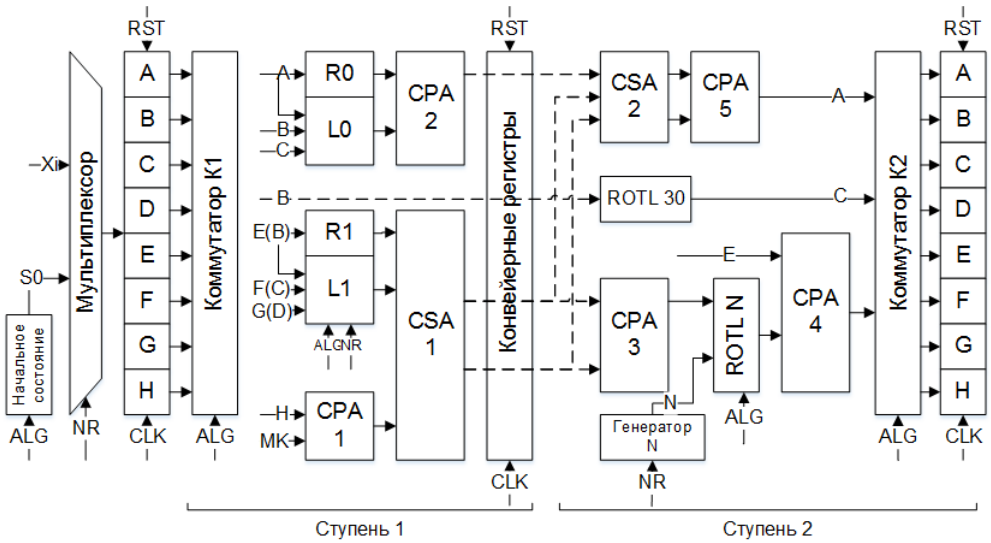

Рис 3. Микроархитектура стадии КВХ

При выполнении алгоритмов MD5, SHA-1 производится реконфигурация стадий КВХ в соответствии с кодом ALG. При этом для снижения энергопотребления верхняя вычислительная ветвь

отключается (на ее входы A, B, C подаются нулевые операнды), программируемый логический блок L1 конфигурируется на выполнение функций *L(A,B,C)* в соответствии с кодом ALG и номером раунда NR. При реализации SHA-1 на вход CPA1 вместо *H* подается циклически сдвинутый операнд (*A rotl5*), операнд *B* передается на выход стадии с циклическим сдвигом (*B rotl30*). При реализации MD5 в вычислительную ветвь включается программируемый сдвигатель ROTL N, в котором разрядность сдвига N определяется номерами раунда и итерации.

Мультиплексор на входе КВХ в начальный момент времени (при включении питания или сигнале RST) загружает в регистры исходные значения хеш-операндов C0, которые поступают на обработку. Ввод C0 продолжается, пока с выхода КПД не поступит блок MWi, имеющий признаки V = 1, L = 0, NR = 0 (первый обработанный блок). Начиная с этого такта, в регистры записываются операнды, поступающие с выхода КВХ. При поступлении блока, имеющего признаки V= 1, L = 1, NR = 3 (последний обработанный блок сообщения), мультиплексор снова переходит в режим загрузки начальных значений C0 для обработки блоков нового сообщения.

Конвейер КПД производит передачу между стадиями и преобразование блоков данных MWi (шестнадцать 32-разрядных слов Mi). Каждый блок передается вместе с 32-битным заголовком LBi, формат которого показан на рис. 2. Преобразование слов данных выполняется, начиная с итерации i=16, в соответствии с функциями:

для алгоритма SHA2/256:

$$Mi = M_{i-16} + M_{i-7} + S0(M_{i-15}) + S1(M_{i-2}), \quad (5)$$

$$S0 = (M_{i-15} \text{ rotr } 7) \text{ xor } (M_{i-15} \text{ rotr } 18) \text{ xor } (M_{i-15} \text{ shr } 3),$$

$$S1 = (M_{i-2} \text{ rotr } 17) \text{ xor } (M_{i-2} \text{ rotr } 19) \text{ xor } (M_{i-2} \text{ shr } 10),$$

где *rotr* – циклический сдвиг вправо,

shr - логический сдвиг вправо;

для алгоритма SHA-1:

$$Mi = (M_{i-3} \text{ xor } M_{i-8} \text{ xor } M_{i-14} \text{ xor } M_{i-16}) \text{ rotl } 1. \quad (6)$$

На каждой стадии КПД производится также суммирование слова Mi с константой Ki.

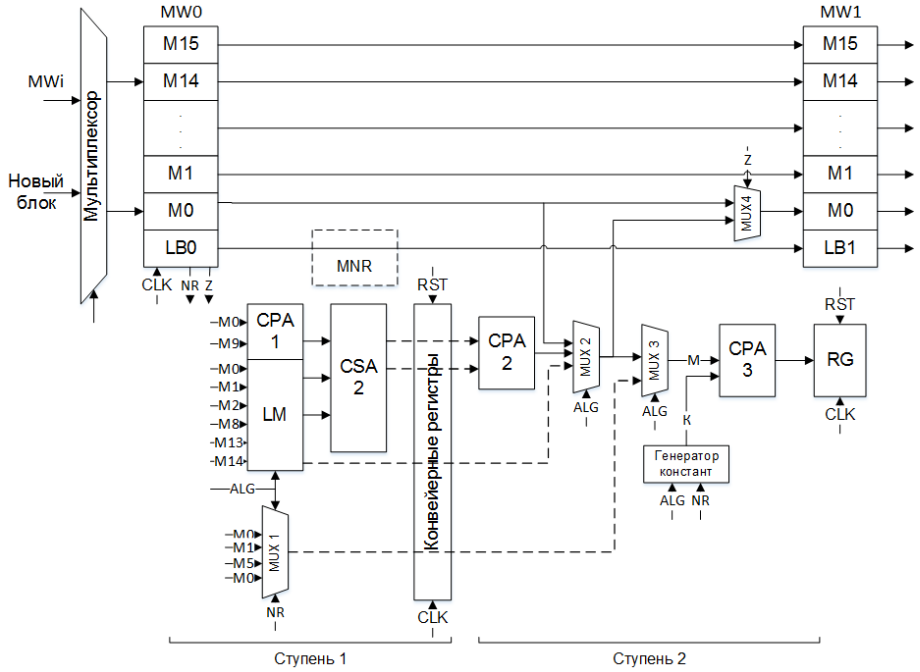

Рис 4. Микроархитектура стадии КПД

Микроархитектура стадии SDi (рис. 4) содержит сумматоры CPA, CSA, мультиплексоры и программируемый логический блок LM, который выполняет операции *S0*, *S1*, а также вычисляет значение *Mi* в соответствии с выражением (6) при реализации SHA-1. При реализации MD5 необходимая перестановка слов осуществляется мультиплексором MUX1. Номера слов, используемых на стадиях КПД, зависят от номера раунда и выполняемой итерации (на рис. 4 дана нумерация слов для стадии SD0). Выбор константы *Ki* зависит от алгоритма, номеров итерации и раунда. Номер раунда NR считывается из заголовка блока LBi и используется на соответствующих стадиях SHi, SDi. В КПД используется также признак Z, который блокирует преобразование слов в первом раунде, обеспечивая снижение энергопотребления.

Для организации длинных раундов (алгоритм SHA-1) в состав стадий SD3, SD7, SD11 включены модификаторы MNR, которые увеличивают на 1 значение NR в заголовке блока, соответственно, после 19-й, 39-й и 59-й итерации. Мультиплексор на входе КПД позволяет загрузить новый блок данных или

продолжить обработку ранее введенных блоков, которые поступают с выхода КПД для выполнения следующих раундов.

Каждая стадия КВХ и КПД делится конвейерными регистрами на две ступени, при этом длительность конвейерного такта равна двум периодам синхронизации криптоблока. Для КВХ длина критической цепи для ступеней 1 и 2 составляет n1 = 12, n2 = 15, для КПД - n1 = 11, n2 = 16. Микроархитектура ступеней КВХ и КПД позволяет выполнять за два периода синхросигнала CLK одну итерацию вычисления хеша согласно заданному алгоритму MD5, SHA-1 или SHA-2/256.

IV. Оценка результатов и выводы

С использованием библиотеки КМОП-элементов с проектными нормами 65 нм проведен логический синтез и оценка характеристик 16-ступенчатых конвейеров и криптоблока, содержащего СК и четыре ИК. Полученные данные (табл. 3) показывают, что пиковая пропускная способность криптоблока достигает 202 Гбит/с для алгоритмов MD5, SHA-2/256 и 162 Гбит/с для алгоритма SHA-1. Разработанные устройства могут использоваться как сложно-функциональные блоки при проектировании высокопроизводительных криптопроцессоров, реализуемых в составе «систем на кристалле».

В таблице 4 приведены характеристики различных криптоблоков, реализованных на базе FPGA, где указаны: S - количество программируемых модулей (slice), F - максимальная рабочая частота, P - пиковая пропускная способность. Для оценки эффективности архитектурных и схемотехнических решений используется отношение пропускной способности к количеству аппаратных ресурсов (число slice) $K = P/S$.

Таблица 3

Характеристики разработанных блоков

	Площадь кристалла	Рабочая частота	Потребляемая мощность	Пропускная способность
КВХ	0,38 мм^2	1000 МГц	271 мВт	-
КПД	0,48 мм^2	1000 МГц	298 мВт	-
СК	0,41 мм^2	1000 МГц	209 мВт	-
ИК	0,87 мм^2	1000 МГц	583 мВт	256 (205) Гбит/с
Криптоблок	4,3 мм^2	790 МГц	1908 мВт	202 (162) Гбит/с

Приведенные данные показывают, что предлагаемая мультиконвейерная архитектура позволяет существенно повысить пиковую пропускную способность криптоблоков, что обеспечивается ценой увеличения используемых аппаратных ресурсов. При этом эффективность данной архитектуры по показателю «производительность / ресурсы» оказывается значительно выше, чем для других вариантов криптоблоков.

Высокая пиковая пропускная способность криптоблоков с макроконвейерной архитектурой реализуется при непрерывной полной загрузке исполнительных конвейеров блоками обрабатываемых сообщений. Поэтому основной областью применения таких криптоблоков являются системы обработки высокоскоростных потоков сообщений.

Таблица 4

Сравнение характеристик

	Семейство FPGA	Реализуемые алгоритмы	S	F, МГц	P, Мбит/с	K
Мультиконвейерный криптоблок	Virtex-7	MD5, SHA-2/256 SHA-1	26 292	128	32 768 26 214	1,25 1,0
Криптоблок [9]	Virtex-II Pro	SHA-2/256	994	100	785	0,79
Криптоблок [10]	Spartan-2	SHA-1	423	106	212	0,50
Криптоблок [5]	Stratix II	MD5, SHA-2 SHA-1	1662	93	721 580	0,43 0,34
Криптоблок [11]	Virtex-II	SHA-2/256	1 260	69	276	021
Криптоблок [12]	Virtex-5	SHA-1	1 351	125	786	0,58

Литература

[1] Rivest R. The MD5 Message-Digest Algorithm // RFC 1321. MIT and RSA Data Security, April 1992.

[2] Federal Information Processing Standards. Secure Hash Standard // FIPS PAB 180-4, 2012.

[3] Lien R., Grembowski T., Gay K. A 1 Gbit/s Partially Unrolled Architecture of Hash Function SHA-1 and SHA-512 // Proceedings CT – RSA. – 2004. – P. 324-328.

[4] Machetti. V., Dadda L. Quasi-pipelined hash circuits // Proceedings of the IEEE Symposium on Computer Architecture. -2005. – P. 317-322.

[5] Ducloyer S., Vaslin R., Cogniat G., Wanderley E. Hardware implication of a multi-mode hash architecture for MD5, SHA-1 and SHA-2 // Workshop on Design and Architectures for Signal and Image Processing . – 2007.

[6] Hoang A.T., Yamazaki K., Oyanagi S. Three-stage pipeline implementation for SHA2 using data forwarding // 2008

International Conference on Field Programmable Logic and Application. - 2008. - P. 29-34.
[7] Rote M.D., Vijendaran N., Selvakumar D. High performance SHA-2 core using the Round Pipelined Technique // 2015 International Conference on Electronics, Computing and Communication Technologies. – 2015. – P. 1-6.
[8] Шагурин И.И., Жихарев Г.Ю. Высокопроизводительные криптоблоки для ипользования в составе «систем на кристалле» // Датчики и системы. – 2016, №4.
[9] Chaves R., Kuzmanov G., Vassiliadis S., Sousa L. Reconfigurable Cryptographic Processor // Workshop on Circuits, Systems and Signal Processing – CSSP'06. - 2006. – P. 1-7.
[10] Wang G. An Efficient Implementation of SHA-1 Hash Function // IEEE International Conference on Electro or Information Technology. – 2006. – P. 575-579
[11] Glabba R., Imbert L., G, Jullien G., Tisserand A., Veryat-Charvillon N. Multi-mode operator for SHA-2 hash functions // Journal of Systems Architecture. – 2007, v.63, №2-3. – P. 127-138.
[12] Iyer N.C., Mandial S. Implementation of Secure Hash Algorithm – 1 using FPGA // International Journal of Information and Computation Technology. – 2012. V. 3. №8. – P. 757-764.

Multi-pipelined architecture of high-performance crypto-blocks for using in "Systems on a Chip"

I.I. Shagurin, G.Yu. Zhikharev

National Research Nuclear University MEPhI, IIShagurin@mephi.ru

***Keywords* — crypto-algorithm, cryptoblock, hash-sum, executing pipeline, multi-pipelined architecture, throughput, system-on-chip (SoC).**

ABSTRACT

Hash-algorithms are used to obtain the fixed-size fingerprint, or hash-sum, of an arbitrary long message. The most important applications for hash-algorithms are message authentication, and the creation of both digital signatures and one-way password files. In recent years, the most widely used hash-algorithms are MD-5, SHA-1 and SHA-2/256 which produce a unique 128, 180 or 256 bit vector respectively. All of them are based on sequential processing of the consecutive blocks of data. The input message is processed in 512-bit blocks (16 words with 32 bits each) and each block is consequentially scheduled [1], [2]. Message scheduling consists of 64 or 80 iterations, which execute addition, shift, rotation or logical operations on 32-bit state variables and block words.

Crypto-blocks are used in a wide range of microcontrollers or as IP-blocks in system-on-chip (SoC) designs to accelerate hash computation. Hardware implementation targets are as ASIC as reconfigurable hardware (FPGA) platforms.

For high-speed message processing applications (High Definition Television, videoconferencing, Virtual Private Networks, etc.) the performance (throughput) of crypto-block plays the crucial role. The efficient performance improvement is pipelining of hash-sum computation paths. In [3] – [7] the different approaches for pipeline organization are described.

In this paper we introduce the multi-pipeline architecture that provides significant increasing in performance of crypto-blocks. The design consists of a system controller (SC) and four ring executing pipelines (REP). Every REP contains two hash evaluating lines each working in parallel: data preparing line (DPL) and hash executing line (HEL). Each line consists of 16 stages producing hash algorithm steps. Every cycle REP receives data word. When full block is loaded, different words are processed on different stages. To calculate hash-sum for one block we need to process 4 (MD5, SHA-2/256) or 5 (SHA-1) full pipeline executing cycles. After block processing was done we replace old data block by a new one.

SC is used for padding messages (creating blocks), distributing blocks between REPs and setting a work mode according to hash-algorithm and round number. Our design uses four REPs, so 64 data blocks can be processed in parallel. When block processing needed 64 iterations (MD5, SHA-2/256), after receiving all 64 data words the first REP finishes the processing of first data block and the next 65-th is loaded. When all REPs were fully loaded, crypto-block is able to process a continuous message stream for hash-algorithms MD5, SHA-2/256 and calculate hash-sums every pipeline cycle. The maximum achieved throughput is $P = 1/T_p$, where T_p is the time for scheduling one iteration on REP stage.

SHA-1 needs 5 full pipeline executing cycles, that's why message processing throughput is 25% less in comparison with other hash-algorithms.

The proposed design is able to execute a single iteration of hash calculating with 2 clock cycles. DPL and HEL consist of registers, carry save adders, logical functions, programmed shifters and multiplexers. Each stage of pipeline is divided into 2 independent steps, so the duration of pipeline cycle T_p is equal to 2 clock periods T_{clk}.

The described circuit was implemented in Verilog and synthesized with CMOS technology library, featuring 65nm silicon process. Synthesis results are: area 4.3 mm^2, power consumption 1,9 W when operating at maximum

frequency F_{clk} = 690 MHz. According to the results, maximum throughput is 202 Gb/s for MD5, SHA-2/256 and 162 Gb/s for SHA-1.

The proposed architecture was also synthesized and implemented on Xilinx Virtex-7 FPGA. The obtained results are: slices S = 26 292, frequency F_{clk} = 128 MHz, throughput P = 32,77 Gbit/s for MD5 and SHA-2/256 and P = 26,20 Gbit/s for SHA-1. Implementation results indicate 30-40 throughput P gain for the proposed crypto-block compared to other designs [9] - [12] implemented on different types of FPGA. This gain is achieved at the cost of increase in hardware resources S to the factor of 16-26. Anyway, the parameter $K = P/S$ for multi-pipelined crypto-block still remains with 1.5 – 2.0 times higher.

The obtained results suggest the following conclusions:

1. The proposed design can be used as IP-block for high performance crypto-processors in system-on-chip design.
2. The maximum throughput of pipeline crypto-block is achieved when pipeline is continuous fully loaded with message stream. So, the main application for such devices are high-speed message flow processing systems.
3. The proposed multi-pipeline architecture allows to significantly increase the maximum throughput at the cost of increasing hardware resources. Along with that the efficiency of this architecture by the "performance/resources" parameter is considerably higher than that of the other examined cryptoblocks.

REFERENCES

[1] Rivest R. The MD5 Message-Digest Algorithm. RFC 1321. MIT and RSA Data Security, April 1992.

[2] Federal Information Processing Standards. Secure Hash Standard. FIPS PAB 180-4, 2012.

[3] Lien R., Grembowski T., Gay K. A 1 Gbit/s Partually Unrolled Architecture of Hash Function SHA-1 and SHA-512. Proceedings CT – RSA. 2004, pp. 324-328.

[4] Machetti. V., Dadda L. Quasi-pipelined hash circuits. Proceedings of the IEEE Symposium on Computer Architecture, 2005, pp. 317-322.

[5] Ducloyer S., Vaslin R., Cogniat G., Wanderley E. Hardware implication of a multi-mode hash architecture for MD5, SHA-1 and SHA-2. Workshop on Design and Architectures for Signal and Image Processing, 2007.

[6] Hoang A.T., Yamazaki K., Oyanagi S. Three-stage pipeline implementation for SHA2 using data forwarding. 2008 International Conference on Field Programmable Logic and Application, 2008, pp. 29-34.

[7] Rote M. D., Vijendaran N., Selvakumar D. High performance SHA-2 core using the Round Pipelined Technique. 2015 International Conference on Electronics, Computing and Communication Technologies, 2015, pp. 1-6.

[8] Shagurin I.I., Zhikharev G.Y. High – performance cryptoblocks for using in "systems on chip". Datchiki I sistemy – Sensors and Systems, 2016, no. 4, pp. (In Russian).

[9] Chaves R., Kuzmanov G., Vassiliadis S., Sousa L. Reconfigurable Cryptographic Processor. Workshop on Circuits, Systems and Signal Processing – CSSP'06, 2006, pp. 1-7.

[10] Wang G. An Efficient Implementation of SHA-1 Hash Function. IEEE International Conference on Electro or Information Technology, 2006, pp. 575-579

[11] Glabba R., Imbert L., G, Jullien G., Tisserand A., Veryat-Charvillon N. Multi-mode operator for SHA-2 hash functions. Journal of Systems Architecture, 2007, v.63, no. 2-3, pp. 127-138.

[12] Iyer N.C., Mandial S. Implementation of Secure Hash Algorithm – 1 using FPGA. International Journal of Information and Computation Technology. 2012. V. 3. no. 8. pp. 757-764.