

РЕАЛИЗАЦИЯ БИБЛИОТЕЧНЫХ ЭЛЕМЕНТОВ ДЛЯ ЦИФРОВЫХ УСТРОЙСТВ НА КРИСТАЛЛАХ ПЛИС ТИПА FPGA

Палагин А.В., Опанасенко В.Н., Сахарин В.Г.

Институт кибернетики НАН Украины

В настоящее время реконфигурируемые аппаратные средства [1] на основе программируемых логических интегральных схем (ПЛИС) становятся реальной и быстро развивающейся областью вычислительной техники. К настоящему времени ПЛИС прошли путь от простых кристаллов ПЛМ до сложных FPGA - программируемых пользователем вентильных матриц с логической емкостью в несколько миллионов вентилей на кристалл. Показатели быстродействия выросли с 20 до 400 МГц. Резко возросли также и показатели конструктивной сложности (по сравнению с используемыми ранее 20 – контактными корпусами) - в современных ПЛИС число внешних выводов увеличилось почти в 60 раз, а плотность упаковки – примерно в миллион раз.

Структурно ПЛИС представляют собой однородную среду и обладают свойствами: однородности, реконфигурируемости, параллельности выполнения операций. Увеличение ресурсов современной программируемой логики и уменьшение их стоимости (в частности ПЛИС типа FPGA) позволяют существенно повысить быстродействие разрабатываемых устройств и аппаратно реализовать алгоритмы, работающие в реальном режиме времени. Распараллеливание вычислений или логических операций может осуществляться как на уровне разрядов представления информации, так и на уровне блоков, выполняющих требуемые алгоритмы математической модели. Другой важной особенностью программируемой логики является свойство реконфигурируемости, т.е. возможность изменения алгоритма работы в зависимости от изменившихся условий или требований. Реконфигурируемость – свойство системы переопределять совокупность аппаратных средств и соединений между ними в соответствии с требуемым алгоритмом функционирования.

При этом предполагается, что модифицированный алгоритм (или ряд алгоритмов), в виде загружаемого в кристалл файла конфи-

гурации, должен быть подготовлен заранее. Файл конфигурации – это программный файл, сформированный посредством САПР для конкретного типа кристалла ПЛИС, предназначенный для создания требуемой конфигурации в кристалле ПЛИС. Процесс проектирования и верификации выполняется средствами САПР. Одним из трудоемких этапов, предшествующих процессу проектирования, является этап постановки задачи, заключающийся в описании алгоритма и его декомпозиции на части, каждая из которых представляет собой алгоритм, имеющий известное аппаратное представление (например, сумматор, умножитель и т.д.). Незначительное время программирования кристалла, исчисляемое в среднем миллисекундами, позволяет осуществлять его динамическое реконфигурирование в процессе выполнения работы. Такая работа, называемая *reconfigurable computing*, позволяет использовать кристалл для выполнения различных задач в режиме разделения времени и приводит к существенной экономии ресурсов.

Основные факторы эффективности применения *reconfigurable computing*: более высокая плотность и быстродействие устройств FPGA, масштабируемость (*scaleable*), реконфигурируемая архитектура компьютера, языки проектирования высокого уровня (для FPGA), простота и доступность средств проектирования.

По существующим прогнозам, ПЛИС серии Virtex, которые начали производиться с 1995 года, за последующие 10 лет увеличат логическую емкость от 50 тыс. до 100 млн. логических вентилей. Если первые кристаллы серии Virtex производились по технологии 0,35 мк, а в настоящее время совместно с фирмой IBM (2002 год) – 0,15 мк, то в 2005 году ожидается производство кристаллов Virtex по технологии 0,07 мк.

В составе современных компьютеров подсистемы на основе FPGA часто используются в качестве сопроцессоров, подключаемых к стандартным шинам типа PCI, CompactPCI, VME. В отличие от традиционной Фон - Неймановской архитектуры, которая обеспечивают общее решение для всех алгоритмов, архитектура сопроцессора, реализованного на FPGA, индивидуальна и выполняет вполне определенный алгоритм, запрограммированный разработчиком. Алгоритм может разбиваться на фрагменты, каждый из которых в виде файла

конфигурации загружается в FPGA и выполняется, реализуя механизм динамической реконфигурации.

Реконфигурируемые устройства в виде плат сопроцессоров для задач телекоммуникации, цифровой обработки сигналов и моделирования специализированных устройств разработаны фирмами Annapolis Micro Systems, Alpha, Nallatech, Aptix и др. В то же время компания Star Bridge Systems [2] представила новый гиперкомпьютер HAL (hyper algorithmic logic), основанный на реконфигурируемой технологии и построенный на 280 кристаллах FPGA фирмы Xilinx. HAL имеет собственную систему программного обеспечения Viva, которая является операционной системой с инструментальным программированием для специального применения. Кристаллы изменяют структуру и функции непрерывно для решения множества вычислительных задач в режиме реального времени.

Для реконфигурируемых вычислительных устройств и систем процесс разработки метода и алгоритма решения исходной задачи носит итеративный характер. Критериями эффективности искомого метода (алгоритма) являются обобщенные характеристики производительности, аппаратных затрат, точности решения задачи, сложности алгоритма, надежности проектируемой системы либо специальные критерии, такие как работа в реальном времени, трудоемкость разработки соответствующих метода и алгоритма и др.

Предлагается создание хорошо структурированной библиотеки методов и соответствующих архитектур проектируемых цифровых устройств, и выбор подходящей пары (метод - архитектура) для конкретной проблемной ситуации. Таким образом, задача оптимального синтеза сводится к задаче оптимального выбора на предварительно сформированном (и постоянно расширяемом) множестве решений. При этом необходимо учитывать наличие весьма мощного ряда кристаллов ПЛИС: каждая из имеющихся серий представлена множеством кристаллов, отличающихся быстродействием, потребляемой мощностью, логической емкостью, типом корпуса, количеством выводов и другими важными параметрами.

Целевая функция в аналитическом виде находится одним из приближенных методов, например линейной или нелинейной интерполяции или экстраполяции, по нескольким опорным точкам (струк-

турные реализации алгоритма), которые получают путем предварительного формирования вариантов реализаций алгоритма, или берут из набора готовых CORE – ядер, которые входят в состав САПР ПЛИС [3]. Из множества этих точек, где каждой r – ой точке ($r = 1 \div m$) соответствует реализация с параметрами $\langle T_r, Q_r \rangle$, формируется множество Парето на плоскости $T - Q$ с учетом соотношений:

$$\begin{aligned} T_1 \leq T_2 \leq \dots \leq T_r \leq \dots \leq T_m ; \\ Q_1 \geq Q_2 \geq \dots \geq Q_r \geq \dots \geq Q_m . \end{aligned} \quad (1)$$

В общем виде, задача выбора оптимального варианта реализации алгоритма сводится к минимизации функционала

$$L_r = \alpha T_r + \beta Q_r \Rightarrow \min \quad (2)$$

с учетом ограничений:

$$\begin{cases} T_r \leq T_0 ; \\ Q_r \leq Q_0 , \end{cases} \quad (3)$$

где: α, β - весовые коэффициенты, которые могут быть определены, например, методом экспертных оценок; T_0, Q_0 - заданные предельные значения параметров T_r и Q_r .

Если заданным ограничениям удовлетворяет единственная точка множества Парето, то реализация, соответствующая этой точке, является результирующей. Если ограничениям удовлетворяют несколько таких точек, то необходимо минимизировать функционал.

Абстрактная архитектура реконфигурируемой платы (РП) может быть описана тройкой:

$$S = \langle A_i, P, F \rangle ,$$

где: $P = \{P_i\}$ - множество методов решения поставленной задачи ($i = 1 \div h$);

$A_i = \{A_{ij}\}$ - множество алгоритмов реализации заданного метода, выполняющих функцию отображения множества входных сигнала

лов $\{X_{ij}\}$ в множество выходных сигналов $\{Y_{ij}\}$ для i -го метода $A_i : X_{ij} \Rightarrow Y_{ij}, (j = 1 \div d)$;

$F = \{F_\gamma\}$ - множество файлов конфигурации ($\gamma = 1 \div k, k = h \times d$), определяющих структурные реализации алгоритмов A_{ij} для методов P_i .

Если при аппаратной реализации алгоритм A_{ij} не удастся разместить в один кристалл FPGA, то этот алгоритм разбивается на фрагменты, выполняемые последовательно. Сложность фрагментов алгоритма при этом определяется лишь логической емкостью кристалла. Соответствующие этим фрагментам файлы конфигурации $F_{\gamma l} (l = 1 \div t, \text{ где } t - \text{число фрагментов алгоритма } A_{ij})$ загружаются в кристалл последовательно. Вариант ($t = 1$) определяет реализацию алгоритма A_{ij} в одном кристалле FPGA.

Объем памяти, необходимый для хранения множества F файлов конфигурации, будет определяться величиной $Q = q \times t \times k$ (q - объем памяти, необходимый для хранения одного файла конфигурации).

Реконфигурируемые платы [4] имеют универсальную матрицу обрабатывающего поля (МОП) заданной размерности, которая конфигурируется для выполнения заданного алгоритма или его части. При конвейерном механизме реализации алгоритма в структуру РП вводятся дополнительные матрицы. Структура РП, представленная на рисунке, содержит s матриц МОП, канал ввода-вывода (КВВ), запоминающее устройство (ЗУ) файлов конфигурации, ОЗУ данных, устройство управления (УУ), шины данных (ШД) и управления (ШУ).

Конвейерный механизм предполагает загрузку файла конфигурации в очередную матрицу параллельно с обработкой данных в текущей матрице. Формат файла конфигурации является стандартным для FPGA и содержит информацию о конфигурации матрицы, т.е. формирует соответствующую принципиальную электрическую схему, реализующую заданный алгоритм.

Матрица МОП представляет собой матрицу универсальных элементов, которым под управлением файла конфигурации F_γ назначается непосредственная функция и формируется структура связей ме-

жду ними. Файлы конфигурации F_γ записываются в матрицу МОП из запоминающего устройства файлов конфигурации ЗУ под управлением УУ. В матрицу МОП по шине ШД могут поступать информационные данные из ОЗУ либо внешние входные данные через КВВ. Результаты обработки из матрицы МОП могут передаваться в канал КВВ как внешние выходные данные или в ОЗУ как промежуточные результаты. Внешние данные могут передаваться также в ОЗУ через канал КВВ под управлением УУ. Множество файлов конфигурации $F = \{F_\gamma\}$ записывается в ЗУ файлов конфигурации через канал КВВ под управлением УУ, который формирует последовательность адресов и управляющих сигналов для ЗУ файлов конфигурации. В дальнейшем устройство управления через шину ШУ управляет работой всего устройства и передачей данных.

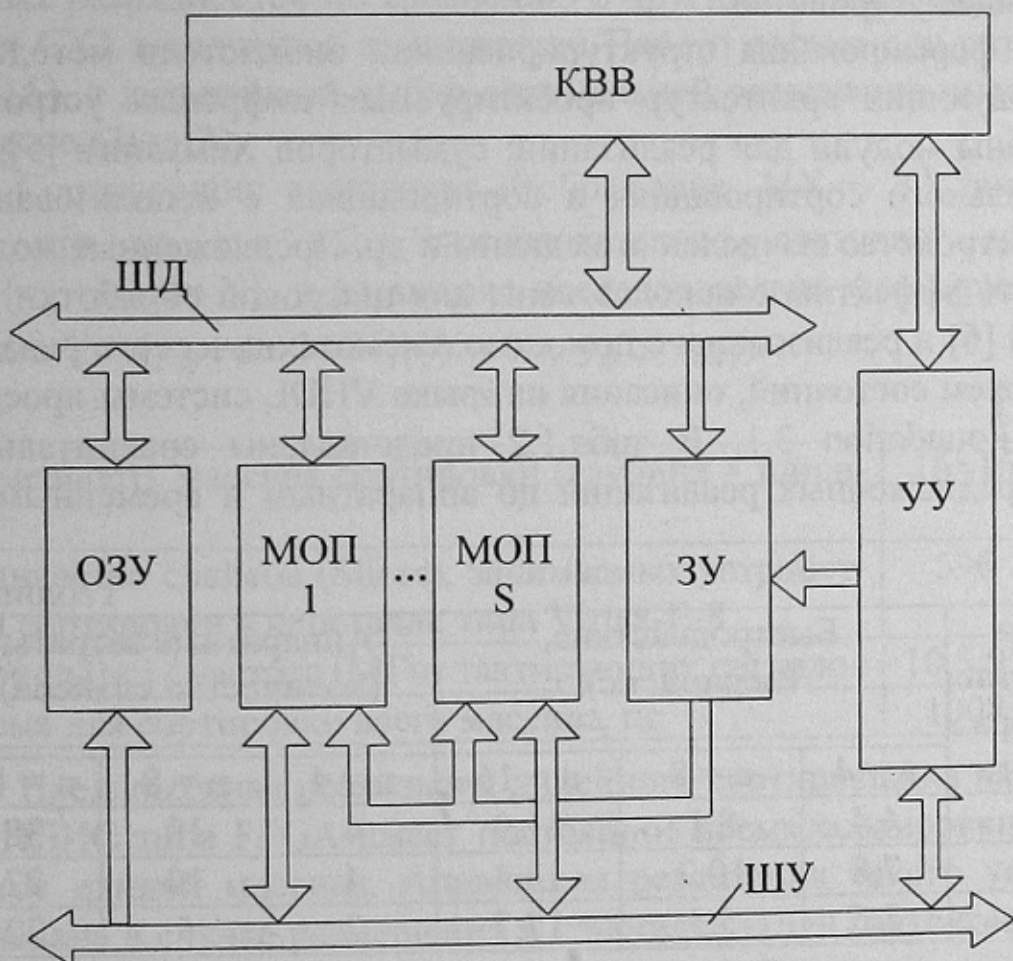


Рисунок 1 - Структурная схема РП

Инициализация работы состоит из трех этапов: запись множества файлов конфигурации F в ЗУ файлов конфигурации; загрузка файлов конфигурации F_γ в МОП из ЗУ файлов конфигурации; функционирование системы – реализация алгоритма.

Последовательность загрузки файлов конфигурации F_γ в матрицы МОП циклическая - $(s + 1) = 1$. Запоминающее устройство файлов конфигурации представляет собой библиотеку структур для обработки данных, записанных в произвольном порядке. УУ посредством управляющего сигнала инициализирует работу соответствующей матрицы МОП.

После загрузки из ЗУ файла конфигурации в МОП будет сформирована структура устройства для реализации соответствующего алгоритма, включающего в свой состав операционное и внутреннее управляющее устройство.

Для формирования структурированной библиотеки методов и соответствующих архитектур проектируемых цифровых устройств разработаны модули для реализации: сумматоров Хемминга [5], последовательного сортировщика и сортировщика с использованием памяти, устройства вычисления медианы и др. Предложенные модули могут быть эффективно использованы для цифровой обработки изображений [6] и реализованы с помощью: схематехнического редактора, граф-схем состояний, описания на языке VHDL системы проектирования Foundation 3.1. В табл.1,2 представлены сравнительные оценки предложенных реализаций по аппаратным и временным затратам.

Таблица 1.

| Вариант реализации сумматора Хемминга | Быстродействие, (период, нс) | | | Аппаратные затраты (количество слайсов) | | |
|---------------------------------------|------------------------------|-------|--------|---|-------|--------|
| | n = 4 | n = 8 | n = 16 | n = 4 | n = 8 | n = 16 |
| CX1 | 4,4 | 8,1 | 11,9 | 4 | 10 | 38 |
| CX2 | 7,8 | 10,2 | 13,4 | 4 | 10 | 22 |
| CX3 | 3,6 | 5,5 | 12,7 | 2 | 5 | 22 |

Сумматор Хемминга представлен следующими вариантами ПЛИС - реализации: CX1 - реализация многоуровневой комбинаци-

онной схемы на основе логических элементов AND, XOR путем схемотехнического ввода проекта; CX2 – реализация на основе сумматоров, сформированных посредством системы Core Generator, и схемотехнического ввода проекта; CX3 – реализация сумматора путем поведенческого описания на языке VHDL.

При формировании множества Парето на плоскости T-Q в соответствии с (1): для ($n = 4$) имеем единственную точку, соответствующую реализации CX3 – $\langle T_1 = 3,6; Q_1 = 2 \rangle$; для ($n = 8$) имеем также единственную точку, соответствующую реализации CX3 – $\langle T_1 = 5,5; Q_1 = 5 \rangle$; для ($n = 16$) получаем две точки, соответствующие реализации CX1 ($\langle T_1 = 11,9; Q_1 = 38 \rangle$) и CX3 ($\langle T_2 = 12,7; Q_2 = 22 \rangle$). Таким образом, реализация CX2 не является оптимальной по временным и аппаратным затратам для рассматриваемых разрядностей по сравнению с другими реализациями. Реализация CX1 включается в множество Парето только для разрядности ($n = 16$), а дальнейший выбор оптимальной реализации определяется согласно (2) и (3).

Сортировщик выполняет отображение $\Lambda(X \Rightarrow Y)$ таким образом, что на его вход последовательно поступает множество $X = \{x_\omega : \omega = 1 \div N\}$, а на выходе последовательно формируется множество $Y = \{y_\varphi : \varphi = 1 \div N\}$ ($Y = X$, $y_\varphi \leq y_{\varphi+1}$).

Таблица 2.

| | | |
|--|----------------|---------------|
| Размерность массива сортировки (глубина \times ширина) | 16 \times 16 | 8 \times 16 |
| Количество слайсов (Slices), занимаемых устройством сортировки в кристалле типа Virtex-E-8 | 596 | 313 |
| Период (нс) / частота (МГц) тактирующих сигналов | 10,5/95 | 9/111 |
| Время для сортировки всего массива, нс | 1008 | 432 |

Предложенная реализация линейного сортировщика на кристаллах ПЛИС типа FPGA имеет постоянное время сортировки, определяемое длиной массива. Аппаратная реализация такого устройства оправдана в случае размещения в кристалле ПЛИС автономного устройства, либо в совокупности с другими устройствами.

Наличие внутренней памяти в кристаллах ПЛИС типа Virtex, в рамках ограничений на имеющийся объем памяти, позволяет реализовать алгоритм сортировки, основанной на использовании оперативной памяти (RAM). Объем RAM зависит от объема сортируемого массива, а затрачиваемые логические ресурсы блока управления – соответственно от объема RAM. Разработанное устройство реализовано в кристалле XCV50E-8 (наименьший по объему кристалл серии Virtex-E) и включает в себя память объемом 2048 8-разрядных слов, позволяющую производить сортировку массива кодов в пределах от 0 до 2047. При этом число повторений любого из кодов в сортируемом массиве не должно превышать 255. Используемые ресурсы: 4 блока из имеющихся 16 блоков RAM (25%) и 50 из имеющихся 768 слайсов (6,5%).

Период синхросигналов CLK составляет 10 нс, цикл записи одного кода в RAM равен соответственно $10 \times 4 = 40$ нс, а полного массива около 82 мкс. Время, занимаемое полным процессом сортировки, составляет около 170 мкс.

Реконфигурируемая плата сопроцессора, при наличии структурированной библиотеки файлов конфигураций, позволяет решать практически любую задачу цифровой обработки сигналов, начиная от предварительной фильтрации поступающего сигнала и его ввода в компьютер до построения сложных систем спектрального анализа, сопроцессоров DSP и др. Актуальность предложенного подхода к построению реконфигурируемых систем возрастает с увеличением размерности функционального поля ПЛИС и сложности реализуемых алгоритмов в особенности для приложений реального времени.

Список источников:

1. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Реконфигурируемые структуры на ПЛИС // УСиМ. – 2000. – №3. – С. 32 – 39.
2. Available at <http://www.starbridgesystem.com/>.
3. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Особенности проектирования цифровых устройств на современных ПЛИС фирмы Xilinx // Проблемы управления и информатики. – 2001. – №1. – С. 105 – 119.
4. Реконфигурируемый процессор. Патент України № 34876 А / 15.03.2001. Бюл. № 2. // Палагин О.В., Опанасенко В.М., Сахарин В.Г.
5. Опанасенко В.М., Сахарин В.Г. Реалізація суматора Хемінга в елементному базисі ПЛИС // Науково – технічна інформація. – 2002. – №1. – С 35 – 38.