

УДК 004.2 (045)

з 943.20-02-5-05 +
з 844.150.2-02-5-05Палагин А. В. чл.-кор. НАН України
Опанасенко В. Н., канд. техн. наук
Сахарин В. Г. канд. техн. наук

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С РЕКОНФИГУРИРУЕМОЙ (ПРОГРАММИРУЕМОЙ) АРХИТЕКТУРОЙ

Институт кибернетики им. В. М. Глушкова НАН Украины

Рассмотрены принципы построения вычислительных систем с реконфигурируемой архитектурой на базе ПЛИС. Эти системы используются для реконфигурируемой обработки данных. Рассмотрены принципы организации реконфигурации систем на базе ПЛИС, представленные комбинацией фиксированных и переменных аппаратных средств. Предложена структурная организация реконфигурируемого процессора. Определены основные цели в области «Reconfigurable Computing».

Введение

Реконфигурируемые системы на базе программируемых логических интегральных схем (ПЛИС) находят все более широкое применение при создании высокопроизводительных устройств обработки данных, цифровой обработки сигналов, поддержки телекоммуникаций, сопряжения комплексов и систем и т. д.

В отличие от традиционной архитектуры систем с программной интерпретацией алгоритмов, системы с использованием ПЛИС реализуют полностью аппаратную или смешанную – программно-аппаратную интерпретацию [1]. Повышение производительности таких систем обеспечивается как за счет указанной реализации алгоритмов, так и за счет высокой степени параллелизма при выполнении задачи.

Структура ПЛИС представляют собой регулярную среду, обладающую свойствами многократной реконфигурации. Логическая емкость современных ПЛИС обеспечивает возможность создания в одном кристалле большого числа аппаратных фрагментов, реализующих заданные алгоритмы. Высокая степень параллелизма достигается за счет одновременного выполнения алгоритмов, а многократная (динамическая) реконфигурация всего кристалла или его части в процессе выполнения задачи позволяет

реализовать многофункциональные или сложные системы с существенной экономией аппаратных ресурсов.

К другим достоинствам ПЛИС следует отнести:

- универсальность, т.е. возможность создания практически любого цифрового устройства в кристалле при наличии персонального компьютера и соответствующих инструментальных средств (САПР) [2, 3];
- возможность модификации проектов на любых стадиях разработки и в процессе эксплуатации;
- высокое быстродействие, малую потребляемую мощность и высокую надежность, обеспечиваемые технологией изготовления кристаллов;
- совместимость с окружающей средой за счет возможности выбора уровня напряжения питания и параметров сигналов ввода/вывода;
- низкую по сравнению с заказными и полужаказными СБИС стоимость реализации проектов за счет массового производства кристаллов с регулярной структурой и небольшого времени, затрачиваемого на разработку проектов и их верификацию.

Развитие ПЛИС осуществлялось от простых ПЛИМ (программируемых логических матриц) к сложным – CPLD (Complex Programmable Logic Devices) и

FPGA (Field Programmable Gate Array). В настоящее время производятся кристаллы *Virtex-II Pro* фирмы *Xilinx* емкостью до 10 млн. системных вентиляей, обладающие помимо большой логической емкости новыми функциональными возможностями и более высоким быстродействием по сравнению с кристаллами *FPGA* предыдущих поколений [4]. В течение ближайших трех-четырёх лет прогнозируется выпуск устройств с емкостью 50 млн. системных вентиляей, достаточных для формирования сложных и высокоэффективных систем на одном кристалле.

В состав САПР ПЛИС входит инструментальное средство *CORE Generator*, формирующее ряд типовых технических решений – *Intellectual Properties (IP-Core)* в виде параметрических, т.е. настраиваемых пользователем «ядер», оптимизированных для разных типов ПЛИС [5]. Использование *IP-Core* сокращает время разработки проектов и улучшает их качество. При необходимости проектировщик по сети *Internet* может заказать и оперативно получить, в соответствии со сформулированными техническими требованиями, необходимое ядро и включить его в свой проект.

Помимо увеличения логических ресурсов и повышения быстродействия за счет совершенствования архитектурных и технологических решений общую тенденцию развития ПЛИС определяют два фактора, обеспечивающих как повышение производительности, так и расширение возможностей применения. К этим факторам относятся: встраивание в кристалл быстродействующих модулей общего назначения и элементов, обеспечивающих реализацию стандартов ввода-вывода для осуществления взаимодействия с различными внешними устройствами.

В модули общего назначения кроме блочной *RAM* (в сериях *Virtex-II* и *Virtex-II Pro* каждый блок имеет объем 18 Кб) входят блоки умножения 18×18 бит, а в серию *Virtex-II Pro* дополнительно введены встраиваемые приемопередатчики (до 24) со скоростью передачи от 622 Мбит/с

до 3.125 Гбит/с на канал в дуплексном режиме, а также *RISC*-процессоры *PowerPC* (максимальное количество – 4).

Максимальная физическая дуплексная скорость передачи данных для приемопередатчика (24 канала) составляет 120 Гбит/с.

Ядро *PowerPC* имеет низкую потребляемую мощность 0.9 мВт/МГц и базируется на гарвардской архитектуре с частотой 300 МГц. Ядро содержит 5-ступенчатый конвейер обработки данных, устройство аппаратного умножения и деления, тридцать два 32-разрядных регистра общего назначения, двунаправленный модульно-ассоциативный кэш команд (16 Кбайт) и такой же кэш данных, устройство управления памятью. Реализуется поддержка *IBM CoreConnect* шинной архитектуры, отладки и трассировки программ.

Для взаимодействия с внешними устройствами, т.е. согласования логических сигналов по уровню, используются различные напряжения питания логического ядра и блоков ввода-вывода (*I/O*). Фирма *Xilinx* использует технологию *Select I/O*, позволяющую каждый блок ввода-вывода запрограммировать в соответствии с определенным стандартом ввода-вывода. Серия *Virtex* поддерживает 16 интерфейсных стандартов ввода-вывода, *Virtex-E* и *Virtex-EM* – 20, *Virtex-II* – 25, *Virtex-II Pro* – 22. Все серии обеспечивают совместимость со стандартом шины *PCI*. Имеется возможность программирования выходного тока *I/O* при выборе низковольтных стандартов (*LVTTL*, *LVC MOS*) в пределах от 2 до 24 мА, в сериях *Virtex-II* и *Virtex-II Pro* введено цифровое управление импедансом.

Современные кристаллы ПЛИС включают в свой состав увеличенную локальную память, специализированные умножители и *RISC*-процессоры (*Power PC*). Скорее всего, кристаллы ПЛИС никогда не будут заменять микропроцессоры для вычислительных задач общего назначения, но концепция конфигурируемой обработки данных, вероятно, будет играть возрастающую роль в развитии

быстродействующих реконфигурируемых вычислительных систем. Аналогично с компьютерами, связанными с сетью *Internet*, которые могут автоматически загружать компоненты программного обеспечения для выполнения специфических задач, реконфигурируемые устройства могут, при необходимости, загружать новые конфигурации аппаратных средств, используя технологию *IRL*.

Для проектирования цифровых устройств на базе ПЛИС широко используются языки описания аппаратных средств – *Hardware Description Language (HDL)*, среди которых наибольшую популярность получили *VHDL* [6], *Verilog* и *Abel*.

Барьеры между аппаратными и программными средствами начинают стираться, так как программное обеспечение позволяет конфигурировать аппаратные средства во время его работы. Такая технология разработки объектных аппаратных средств предоставляет проектировщику возможность работы с проектами, написанными на стандартном языке программирования *C*, и загрузки этих проектов соответствующей прикладной программой, используемой как функция языка *C* [7]. Уникальное свойство реконфигурируемости позволяет в реальном времени эффективно выполнять отладку цифровых проектов.

Поэтому создание перспективных средств вычислительной техники, не привязанных к технологическим линиям производства, с возможностью реконфигурируемости, на основе современных кристаллов ПЛИС представляет актуальную научно-техническую проблему.

Reconfigurable Computing

Быстрое развитие современных технологий и производство ПЛИС высокой степени интеграции привело к созданию новых направлений в «*Computer Science*» – «*Reconfigurable Computing*» и «*IRL-Technology*». Термин «*Reconfigurable Computing*» в общем случае обозначает понятие реконфигурируемости структуры компьютера (*hardware*) и, соответственно, изменения процесса обработки данных, вы-

полняемого компьютером. Технология *IRL (Internet Reconfigurable Logic)* предусматривает возможность реконфигурации (в том числе дистанционной, через сеть *Internet*) структур вычислительных устройств, включенных в эту сеть и реализованных на элементной базе ПЛИС типа *FPGA*.

Теоретические основы и примеры практической реализации ЭВМ с гибкой (программируемой) архитектурой, в которой использовался механизм микропрограммной эмуляции как инструмент перестройки архитектуры, прежде всего процессора, рассмотрены в работах [8, 9]. Важным качеством ЭВМ с гибкой архитектурой является степень гибкости или уровень программируемых компонентов. Именно они определяют диапазон технических решений и свойства архитектур, каждая из которых эффективна в своем, вполне определенном, классе задач. В данном случае уровень программируемых компонентов опускался до функциональных узлов системы управления ЭВМ, поэтому условно его можно назвать «автоматным» уровнем. Базовыми компонентами на то время были программируемые логические матрицы ПЛМ.

С появлением современных кристаллов ПЛИС типа *FPGA* с емкостью более чем 100.000 логических вентилях стало возможным использовать эти кристаллы для построения реконфигурируемых компьютеров с полностью программируемой архитектурой. В реконфигурируемых компьютерах (компьютерах с программируемой архитектурой) фиксируется обрабатываемое поле заданной размерности, сконфигурированное специально для выполнения определенного заданного алгоритма или его части, обеспечивая таким образом реализацию этого алгоритма оптимальным способом, имея в виду, как время его выполнения, так и затраты аппаратных ресурсов. Алгоритм может быть разбит на фрагменты, выполняемые последовательно, в связи с чем, соответствующие этим фрагментам структуры также загружаются в кристалл последовательно (в порядке их выполне-

ния), что приводит к существенной экономии ресурсов. Сложность фрагментов алгоритма при этом определяется лишь логической емкостью кристалла, т.е. размерностью обрабатываемого поля.

Таким образом, реконфигурируемая обработка данных представляет собой в известной мере изменение центральной парадигмы проектирования современных средств вычислительной техники и автоматики, а реконфигурируемые аппаратные средства становятся реальной и быстро развивающейся областью вычислительной техники. Процесс конфигурации ПЛИС, составляющих основу реконфигурируемых устройств, может быть реализован при наличии соответствующих файлов конфигурации, полученных в процессе создания проекта с помощью САПР.

Для реконфигурируемых вычислительных 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, а также 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.

Целевая функция в аналитическом виде находится одним из приближенных методов, например линейной или нелинейной интерполяции или экстраполяции, по нескольким опорным точкам (структурные реализации алгоритма), которые по-

лучают путем предварительного формирования вариантов реализаций алгоритма, или берут из набора готовых *IP-Core*. 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:

$$T_1 \leq T_2 \leq \dots \leq T_r \leq \dots \leq T_m;$$

$$Q_{1 \geq} Q_2 \geq \dots \geq Q_r \geq \dots \geq Q_m.$$

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

$$L_r = \alpha T_r + \beta Q_r \Rightarrow \min.$$

с учетом ограничений:

$$\begin{cases} T_r \leq T_0; \\ Q_r \leq Q_0, \end{cases}$$

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.

Принципы построения реконфигурируемых систем

В настоящее время ведутся работы по созданию компьютерной системы с реконфигурируемой (программируемой) архитектурой [10, 11], которая представляет собой проблемно-ориентированные конфигурации применительно к каждой конкретной задаче. Структура реконфигурируемой системы и состоит из двух частей: постоянной (или «фиксированной») части F – *host* компьютера и переменной части V – так называемого «реконфигурируемого» оборудования, которое можно объединять в различные конфигурации (рис. 1).

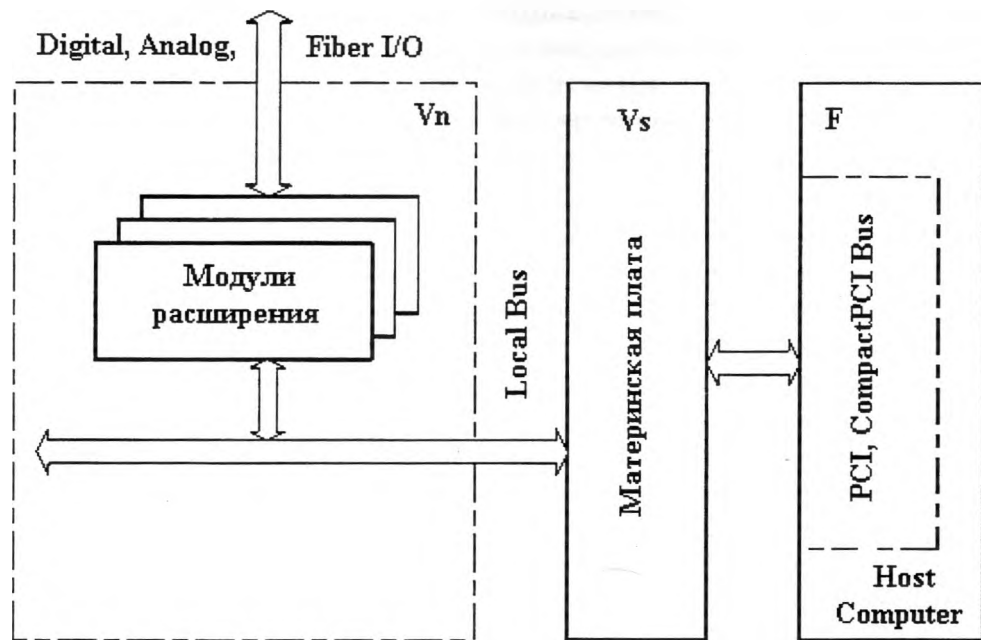


Рис. 1. Структура реконфигурируемой системы

Оборудование V также подразделяется на две части: «стандартную» часть V_s , которая подключается к F через стандартные шины *host* компьютера и представлена материнской платой с локальной внутренней шиной для подключения «нестандартной» части V_n , представляющей собой широкий набор модулей расширения. Операции, выполняемые в каждой из частей, определяются следующими характеристиками: в F – временем вычислений и исходными данными; в V_s и V_n – также дополнительным оборудованием, необходимым для выполнения соответствующих операций, временем передачи информации между вычислительными модулями и временем реконфигурации системы (загрузки *IP-Core* в кристаллы ПЛИС).

В данной системе конфигурация формируется таким образом, чтобы перенести основную работу с F -части системы на специализированные блоки (V -часть), которые представляют собой *soft cores*. Для строго сформулированной вычислительной задачи (где однозначно определены все численные процедуры) и описания характеристик операций для F и V требуется организовать общую структуру

($F + V_s + V_n$) и распределить вычисления таким образом, чтобы минимизировать целевую функцию (сумма стоимостей реконфигурации и времени вычислений).

Данная проблема является исключительно сложной, по существу это комбинаторная задача оптимального синтеза. Ограничение, накладываемое конечным объемом реконфигурируемого оборудования ($V_s + V_n$), не позволяет получить, приемлемый для практики, однозначный метод нахождения оптимального решения. Поэтому находится решение (близкое к оптимальному) методом последовательных приближений.

Реконфигурация структуры включает два этапа. На первом изменяется лишь часть V_s , т.е. механическое изменение не допускается. Если не достигнут заданный критерий оптимизации, то есть часть V_s не имеет достаточной логической мощности и памяти, либо специфических средств ввода-вывода информации, то осуществляется переход ко второму этапу. И дальнейшая реконфигурация системы выполняется также и механическим способом (путем установки модулей расширения в соответствующие разъемы – слоты локальной шины материнской платы).

Структурная организация реконфигурируемых процессоров

Реконфигурируемые процессоры (РП) представляют собой в минимальной конфигурации печатную плату с размещенными на ней одним или несколькими кристаллами пользовательских ПЛИС (ППЛИС), энергонезависимой памяти для хранения файлов конфигурации, элементами загрузки конфигурационного файла (файлов) и одним или несколькими разъемами для подключения внешних устройств (модулей расширения). Тип энергонезависимой памяти определяется областью применения РП: для динамического конфигурирования ППЛИС в процессе функционирования целесообразно использовать *Flash*-память, а при отсутствии такой необходимости – ППЗУ. Использование *Flash*-памяти предполагает наличие в составе РП блока управления этой памятью, реализующего загрузку этой памяти файлами конфигурации из внешнего источника, а также чтения с произвольной выборкой требуемого файла и его загрузки.

Большинство задач обработки данных требуют наличия *Cache*-памяти в составе РП. Обращение к памяти должно осуществляться как от внешнего устройства (через установленный разъем), так и со стороны устройства, реализованного в ППЛИС. С целью расширения памяти на плате РП устанавливается разъем для подключения дополнительной памяти. *Cache*-память в составе РП предполагает наличие контроллера памяти.

Так как в настоящее время наибольшее распространение получила шина *PCI* и реализация контроллера этой шины для РП предлагается в качестве «ядра» как фирмой *Xilinx*, так и другими фирмами, то целесообразно конструкторское исполнение РП в виде платы сопроцессора, подключаемого в слот расширения шины *PCI* базового компьютера. Контроллер шины может быть помещен как в отдельный кристалл ПЛИС, так и в ППЛИС. В последнем случае в комплект поставки РП также должны входить фай-

лы описания контроллера *PCI*, включаемые пользователем в проект ППЛИС в качестве макроэлемента, а файлы конфигурации должны храниться в энергонезависимой памяти. Для загрузки файлов в ППЛИС из *Flash*-памяти требуется контроллер, располагаемый в отдельном энергонезависимом кристалле. Если контроллер шины размещается в отдельном кристалле ПЛИС, то файлы конфигурации, загружаемые в ППЛИС, могут храниться на *hard*-диске (*HD*). Контроллер загрузки файлов целесообразно расположить в кристалле, предназначенном для реализации контроллера шины. В комплект поставки РП в этом случае должна входить программа, выполняющая загрузку конфигурационных файлов в ППЛИС с *HD* через *PCI*-интерфейс.

Такой тип РП конструктивно реализован в виде несущих или материнских плат, в связи с тем, что к ним могут быть подсоединены платы расширения. РП разрабатываются под промышленные стандарты, такие как *PCI*, *CompactPCI*, *PMC* (*PCI Mezzanine Cards*), *DIME* (*DSP and Image processing Module for Enhanced FPGAs*) и *VME*. Спецификация *PMC* разрешает добавлять модули расширения к материнским платам через локальную шину *PCI*. Материнские платы подключаются к стандартной шине компьютера и работают в режиме сопроцессора.

Абстрактная архитектура реконфигурируемого процессора (РП) может быть описана тройкой [12]:

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

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 сигналов $\{X_{ij}\}$ 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 ($\gamma = 1 \div k, k = h \times d$), определя-

ющих структурные реализации алгоритмов A_j для методов P_i .

Если при аппаратной реализации алгоритм A_j 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 $F_{j,l}$ ($l=1 \div t$, где t – число фрагментов алгоритма A_j) загружаются в кристалл последовательно. Вариант ($t=1$) определяет реализацию алгоритма A_j 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 процессоры имеют универсальную матрицу обрабатываемого поля (МОП) заданной размерности, которая конфигурируется для выполнения заданного алгоритма или его части. При конвейерном механизме реализации алгоритма в структуру РП вводятся дополнительные матрицы. 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 между ними. Файлы конфигурации F_{γ} записываются в матрицу МОП из запоминающего устройства файлов конфигурации (ЗУ) под управлением устройства

управления (УУ). В матрицу МОП по шине данных могут поступать информационные данные из ОЗУ либо внешние входные данные через канал ввода/вывода. Результаты обработки из матрицы МОП могут передаваться в канал как внешние выходные данные или в ОЗУ как промежуточные результаты. Внешние данные могут передаваться также в ОЗУ через канал под управлением УУ. Множество файлов конфигурации $F = \{F_{\gamma}\}$ записывается в ЗУ файлов конфигурации через канал под управлением УУ, который формирует последовательность адресов и управляющих сигналов для ЗУ файлов конфигурации. В дальнейшем устройство управления через шину управления координирует работу всего устройства и передачу данных.

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.

Основные задачи в области Reconfigurable Computing

Комплекс основных задач, решаемых в данной области, можно сформулировать следующим образом:

- разработать теоретические основы принципов построения реконфигурируемых цифровых структур на базе кристал-

лов ПЛИС, в соответствии с этим построить систему формализованных методик и алгоритмов синтеза параметрических модулей с учетом особенностей их конструктивно-технологической базы;

- разработать основы теории адаптивных логических сетей, которые предназначены для решения широкого класса задач путем непосредственной структурной реализации алгоритмов обработки. В основу структурной организации таких сетей положены требования динамической реконфигурируемости, многоуровневости и параллельности обработки данных, которым в полной мере соответствует современная элементная база – ПЛИС;

- разработать базовые библиотечные параметрические модули посредством их описания на языке *HDL*.

Обобщенная структура вычислительной системы с реконфигурируемой (программируемой) архитектурой (*ВСПА*) приведена на рис. 2 и содержит компьютер с пакетом проектирования ПЛИС, реконфигурируемое устройство на базе кристаллов ПЛИС и открытую библиотеку файлов конфигурации.

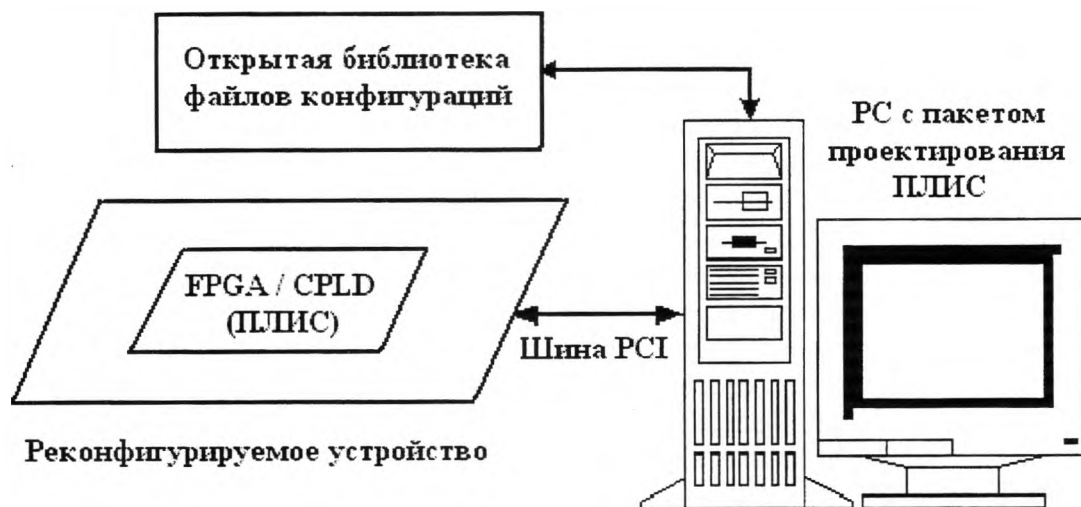


Рис. 2. Обобщенная структура *ВСПА*

Выводы

Создание реконфигурируемых устройств на основе современных кристаллов ПЛИС для вычислительных систем с возможностью реконфигурируемости при наличии открытой библиотеки файлов конфигурации позволит за короткий про-

Принципиальными отличиями *ВСПА* являются:

- ориентация на технологические возможности отечественных предприятий компьютерного машиностроения;

- возможность использования стандартного компьютера PC в качестве основы при построении сложной вычислительной системы;

- живучесть системы, обеспечение возможности замены неисправных функциональных узлов путем динамической реконфигурации;

- свойство структурной универсальности (помимо алгоритмической универсальности), позволяющее для каждого алгоритма создавать в универсальном функциональном поле свою структурную схему, которая обеспечивает эквивалентное отображение структурной схемы алгоритма и процессов функционирования, позволяя менять логическую структуру устройства в зависимости от специфики решаемой задачи путем реконфигурирования внутренней структуры.

межутков времени реализовать в кристаллах новейшие разработки.

Объектом исследования в данной области являются вычислительные системы с реконфигурируемой (программируемой) архитектурой, обеспечивающие:

- аппаратную реализацию современных и перспективных средств вычислительной техники, с возможностью реконфигурируемости на основе современных кристаллов ПЛИС;

- поддержку открытой (с возможностью расширения) библиотеки функциональных параметрических модулей реконфигурируемых цифровых устройств с методической, алгоритмической и технологической поддержкой для разработчиков и различных категорий пользователей при решении задач выбора оптимальной аппаратной реализации широкого класса алгоритмов, в том числе знание-ориентированных.

Реконфигурируемые системы найдут широкое применение в следующих приложениях:

- проблемно-ориентированные системы и сопроцессоры;

- системы телекоммуникаций, цифровой обработки сигналов;

- моделирование алгоритмов и проектирование архитектур современных компьютеров, основанных на кристаллах *FPGA* и *ASIC*;

- современные системы управления, связанные с выполнением больших объемов вычислений – управление технологическими процессами, контрольно-измерительной аппаратурой, роботами-манипуляторами, других системах реального времени;

- знание-ориентированные системы – создание автоматных сетей для синтаксически-семантического анализа текстов, в частности, для реализации модели языковой картины мира.

Список литературы

1. Палагин А. В., Опанасенко В. Н., Сахарин В. Г. Реконфигурируемые структуры на ПЛИС // УСиМ. – 2000. – №3. – С. 32-39.

2. Опанасенко В. Н., Сахарин В. Г. ПЛИС типа *FPGA* фирмы *Xilinx*: возмож-

ности, проектирование и применение // Электронные компоненты и системы. – 2003. – №4. – С. 7-11.

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. *Xilinx Virtex-II Series FPGAs*/ Available at <http://www.xilinx.com/products/platform>.

5. *Core Solutions Data Book*. – *Xilinx Inc.* 1998. – 399 p.

6. *VHDL'93*. IEEE Standard VHDL Language Reference Manual. IEEE Std 1076. – 1993.

7. *Schewel J.* A Hardware / Software Co-Design System using Configurable Computing Technology / Available at <http://www.vcc.com/>.

8. Палагин А. В., Рокитский А. Г. МикроЭВМ с гибкой архитектурой // Механизация и автоматизация. – 1983. – № 3. – С. 10-14.

9. Палагин А. В., Кургаев А. Ф., Рокитский А. Г. К системному проектированию и применению микроЭВМ с гибкой архитектурой // УСиМ. – 1984. – №5. – С. 26-31.

10. *Estrin G., Turn R.* Parallel processing in a restructurable computer // IEEE Transaction on Electronic Computers. – EC-12, N.6. – 1963. – P. 747-755.

11. Опанасенко В. Н., Сахарин В. Г. Реконфигурируемые системы на современной элементной базе // Комп'ютерні засоби, мережі та системи: Зб. наук. праць НАН України. – К.: Ін-т кібернетики ім. В. М. Глушкова, 2003. – №2. – С. 25-32.

12. Пат. 34876 А Україна. Реконфигуровний процесор / Палагин О. В., Опанасенко В. М., Сахарин В. Г. – Чинний від 15.03.2001.