

**А. Т. Орлов** – студент кафедры информационных систем

**А. Ю. Сыщиков** – научный руководитель

## ВЫСОКОУРОВНЕВЫЙ СИМУЛЯТОР МНОГОЯДЕРНЫХ СИСТЕМ НА КРИСТАЛЛЕ

В данной статье рассматривается разработанный высокоуровневый симулятор многоядерных систем на кристалле.

Высокоуровневая симуляция – это важная часть процесса разработки программного обеспечения для многоядерных систем на кристалле (СнК). Она позволяет запускать программы и выполнять функциональную отладку на высокоуровневой модели СнК без углубления в детали разнородных ядер СнК: специфическом наборе команд, взаимодействии процессов, спецификации коммуникационных систем и т.д.

В своем непрерывном развитии рынок микроэлектроники постоянно выдвигает все новые и более жесткие требования к появляющимся изделиям. Потребитель хочет получать быстродействующую, надежную и в то же время малогабаритную и мало потребляющую продукцию. Эти противоречивые требования усугубляются тем, что микроэлектронные поколения очень быстро стареют. Поэтому особое внимание уделяется постоянному сокращению времени выхода на рынок новых изделий, а одно-временная разработка аппаратных и программных компонентов нового продукта является способом решить задачу его сокращения времени выхода на рынок.

Высокоуровневый симулятор - это программа, которая позволяет быстро и с минимальными затратами изучить, настроить и протестировать программно-логическую модель реальной системы без ее построения, а также предоставляют пользователю возможность выполнить тестирование и отладку разработанного программного обеспечения на ней.

Высокоуровневый симулятор для многоядерных СнК должен:

- предоставлять возможность моделировать выполнение высокоуровневого представления программного обеспечения (ПО) на модели СнК с произвольной конфигурацией;
- моделировать характеристики аппаратной платформы:
  - задержки при пересылке и приеме пакетов с данными для каждого аппаратного устройства;
  - задержки при передаче пакетов с данными через каналы;
  - задержки выполнения процессов на процессорных элементах;
  - задержки при обращении к основной памяти и т.д.;
- собирать статистическую информацию о процессе моделирования;
- показывать текстовую и графическую информацию после завершения процесса моделирования.

Входными данными для симулятора являются файлы с описанием программы и с описанием аппаратной платформы моделируемой СнК.

Симуляторным движком, который был выбран для высокоуровневого симулятора СнК, является проект DCNSimulator (Digital Communication Network Simulator) [1]. Это кроссплатформенная библиотека, предоставляющая возможности для дизайна, моделирования и анализа дискретных систем. Она разработана на языке C++ и основана на свободно распространяемых библиотеках SystemC, Qt, Qwt.

Программа для СнК (рис. 1) написана на графическом объектно-ориентированном языке VPL [2]. В программе могут находиться следующие типы объектов:

- операторы;
- данные;
- связи;
- структуры (подпрограммы);
- ссылки.

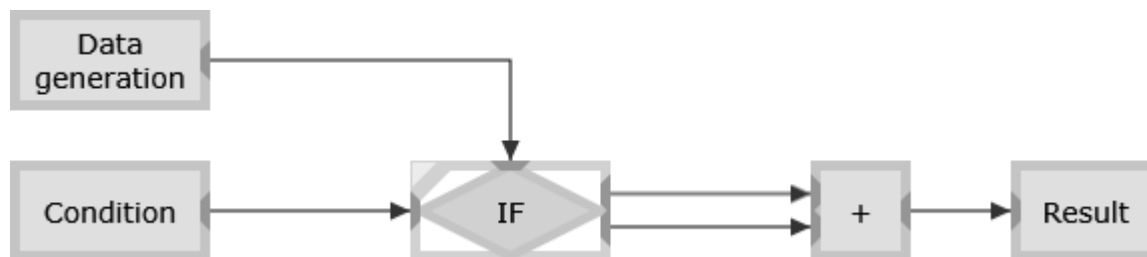


Рис. 1. Пример входной схемы программы

Операторы – это программные объекты, которые выполняют функции обработки данных. Эти функции написаны с использованием специального API на языках программирования С и С++.

Объекты-данные являются пассивными, частично общими хранилищами данных. Модель языка VPL не определяет детали реализации объектов-данных, а определяет только правила функционирования. Поэтому, в представленном симуляторе определена следующая процедура взаимодействия с объектами-данными: если оператор хочет начать работу с объектом-данным, то, сперва, он должен отправить запрос на взаимодействие с объектом-данным или получить разрешение на запись.

Объекты-связи служат для описания соединения программных объектов между собой.

Структуры хранят описание подпрограммы внутри себя. Подпрограмма ничем не отличается от программы, кроме того, что подпрограммы одержат внутри себя объекты-ссылки, которые соединяют их вместе. Каждый объект-ссылка хранит в себе числовое значение, которое указывает на номер порта объекта-структуры, на который он ссылается.

Моделируемая СнК может состоять из процессорных элементов, роутеров и каналов.

Процессорный элемент – это однозадачный вычислительный блок, на котором может располагаться один и более процесс (задача). Процессорный элемент может выполнять только одну задачу в один момент времени, поэтому он содержит очередь на выполнение задач. Более того, переключение задач не допускается, т.е. если задача занимает процессорный элемент вплоть до ее окончания. Процессорный элемент может моделировать свою занятость, например, время задержки выполнения задач. Он содержит входной и выходной буферы данных, через которые он обменивается данными, представленными в виде пакетов, с другими устройствами. Пакеты представляют собой форматированные блоки данных, которые перемещаются по сети. Каждый пакет содержит информации об объекте-отправителе и объекте-получателе, а также данные. После того, как пакет с данными приходит на входной буфер процессорного элемента, он разбирается, и данные из него отправляются на указанный в пакете процесс. Выходной буфер процессорного элемента предназначен для хранения выходных пакетов с данными, которые процессорный элемент не может отправить в канал.

Роутер – это сетевое устройство, которое предназначено для установки маршрутов пакетов с данными через сеть. Каждый роутер содержит столько портов, сколько каналов соединены с ним. Каждый роутер содержит таблицу маршрутизации с привязкой к информации о расположении задач. После того, как роутер принимает пакет с данными, он считывает информацию о пункте назначения пакета и в соответствии с таблицей маршрутизации отправляет этот пакет на соответствующий выходной порт.

Каналы предназначены для соединения различных аппаратных устройств: процессорных элементов с роутерами или напрямую с другими процессорными элементами. Один канал предназначен для соединения двух устройств между собой, т.е. широковестьная передача данных от источника к приемникам по нему запрещена. Канал характеризуется длиной и скоростью передачи данных по нему. Благодаря этим данным моделируется симуляционная задержка времени при прохождении данных через канал. После того, как симулятор получил файлы с программой и описанием конфигурации СнК, он начинает подготовку к моделированию. Сперва, симулятор разбирает файл с программой и собирает информацию об объектах, которые будут участвовать в моделировании. Для этих объектов создаются процессы (задачи). Процесс представляет собой алгоритм выполнения функции определенного типа объекта.

На следующем шаге симулятор разбирает файл с конфигурацией системы на кристалле, создает аппаратные устройства, которые описаны в этом файле, и соединяет их между собой.

Затем созданные процессы равномерно распределяются по созданным процессорным и строятся таблицы маршрутизации для роутеров, если таковые присутствуют в системе.

После окончания процесса моделирования пользователю предоставляется собранная статистическая информация. Информация может быть представлена в текстовом (рис. 2,а) и графическом видах (рис. 2,б).

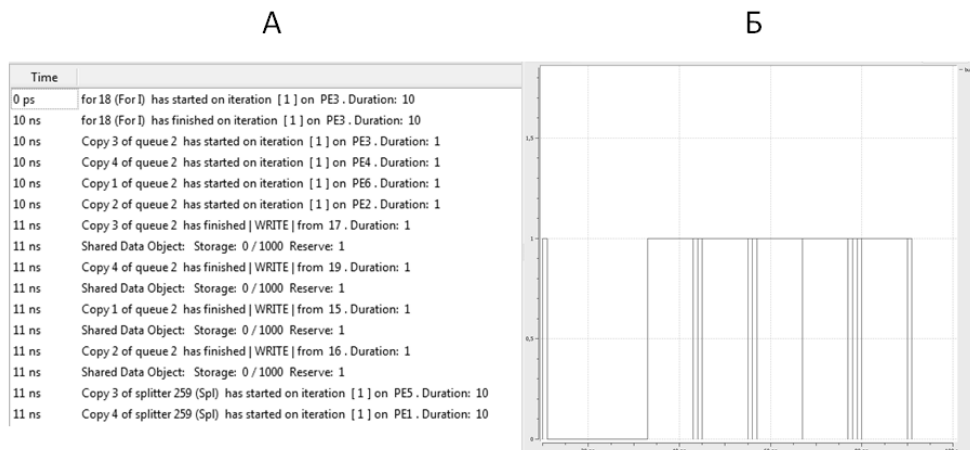


Рис. 2. Статистическая информация

Рассмотренный в данной статье высокоуровневый симулятор многоядерных СнК позволяет выполнять тестирование и отладку ПО для СнК с произвольной конфигурацией, а также анализировать различные характеристики моделируемой системы и разработанного ПО.

#### Библиографический список

1. Sheynin Y.E., Volkov P.L., Onischenko L.V., Razhivin D.B., Cherniy A.S., Eganyan A.V., Nikolsky V.F., Kosyrev S.A., "Software support of the VLSI family "Multicore-designer" for the construction of the parallel structures and distributed signal processing systems", "Questions of Radio electronics", a series of "Electronic computing equipment (EWT)", issue 3, 2008
2. Игнатъев М.Б., Шейнин Ю.Е. и др. Разработка формальных моделей, методов и языка визуального программирования динамических параллельных вычислений в ВС с распределенной архитектурой. Технический отчет. Грант Российского Фонда Фундаментальных Исследований No 01-01-00998, 2003.