

ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ ПО ОРГАНИЗАЦИИ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Россия, г. Пенза, Пензенский государственный технологический университет

The article provides a brief overview of well-known solutions for the organization of reconfigurable computing systems. The relevance of the chosen research topic is due to the ever-growing need to create new methods of organizing computing systems with changing properties (tunable for specific tasks) equipment. In recent decades, there has been significant progress in the field of computer technology. Processor performance continues to grow, which has so far been achieved mainly due to extensive technological development - an increase in parameters such as clock frequencies, the number of instructions executed per clock cycle, cache sizes, the number of cores, etc. At the moment, the increase in the maximum clock frequencies of modern processors has stopped, stopping at the level of 3.5-3.6 GHz, and further development is mainly due to increasing the number of cores and increasing parallelism at the instruction level.

According to Moore's Law (1965), which continues to be true to this day, the number of transistors placed on a chip doubles every 2 years. In the near future, it is possible to reach a certain limit when this law ceases to apply – due to restrictions on reducing physical size, increasing energy consumption, the inability to remove heat, unjustified costs for organizing production and other factors. But so far this process is proceeding exactly at the predicted speed. Reconfigurable systems are designed to fill the existing gap between two diametrically opposite types of computers, the "programmable equipment" of which can change its data processing path, optimizing it for certain applications.

Reconfigurable systems, due to their adaptability and high performance, can make it possible to use the computing resources of modern crystals much more efficiently, increasing the degree of parallelism of calculations for some given tasks, moreover, optimization for each subsequent task does not require an expensive procedure for manufacturing new equipment, but is achieved only by "reprogramming" the reconfigurable unit.

Введение. Цель работы – исследование решений, полезных для создания реконфигурируемых вычислительных систем.

Актуальность темы обусловлена растущей необходимостью создания новых методов организации ВС с изменяющей свойства аппаратурой. В последние десятилетия наблюдался постоянный и значительный прогресс в области вычислительной техники. Производительность процессоров продолжает расти, что до сих пор достигалось, в основном, за счёт технологического экстенсивного развития – увеличения таких параметров, как тактовых частот, количества выполняемых за такт инструкций, размеров кэшей, количества ядер, т.д. На данный момент увеличение максимальных тактовых частот современных процессоров прекратилось, остановившись на величине в районе 3,5-3,6 ГГц, и дальнейшее развитие идёт, в основном, за счёт наращивания количества ядер и увеличения параллелизма на уровне инструкций.

При этом, реальный прирост производительности вычислений не пропорционален увеличению количества транзисторов, и это происходит, в значительной степени, из-за использования в современных процессорах различных вариации фон-неймановской архитектуры, которая имеет некоторые принципиальные недостатки, одним из которых является то, что подобные архитектуры предполагают последовательные алгоритмы

решения задач и малопригодны для параллельных вычислений.

В области встраиваемых систем требования к производительности, стоимости и энергопотреблению обычно гораздо более жёсткие, чем для процессоров общего назначения. В то же время, как правило, решается лишь одна задача, или несколько. Это открывает возможности для специализации и оптимизации, когда спроектированный вычислитель выполняет определённые приложения, для которых предназначен, значительно более эффективно, что достигается за счёт смещения некоторых или даже всех вычислений в аппаратную часть, которая, чаще всего, проектируется вручную, что приводит к большой стоимости и срокам разработки. При этом, большинство систем делается на микроконтроллерах, лишь с незначительным использованием аппаратных ускорителей.

Реконфигурируемые системы, благодаря адаптивности и высокой производительности, могут позволить значительно эффективнее использовать вычислительные ресурсы современных кристаллов, повышая степень параллелизма вычислений для некоторых заданных задач, причём, оптимизация для каждой следующей задачи не требует дорогостоящей процедуры изготовления новой аппаратуры, а достигается всего лишь «перепрограммированием» реконфигурируемого блока. Правда, растёт его конфигурации является процессом не менее трудоёмким, чем проектирование аппаратуры.

Процесс внедрения реконфигурируемых систем осложнён отсутствием единой методологии и инструментов, автоматизирующих процесс создания и программирования реконфигурируемых блоков, в частности, сложность представляет превращение широко используемых сейчас последовательных алгоритмов в параллельные. Поэтому, работы по исследованию методов организации и проектирования реконфигурируемых вычислительных систем на данном этапе развития вычислительной техники широко востребованы. В качестве основы для методологии создания реконфигурируемых вычислителей могут использоваться основы функционирования головного мозга, так как в нём реконфигурируемость присутствует естественным образом, и «отлажена» миллионами лет эволюции биологических организмов.

Реконфигурируемая вычислительная система – вычислительная система, которая может подстраивать свою аппаратную структуру под решаемую задачу с целью оптимизации использования ресурсов (повышая производительность, снижая энергопотребление, т.д.). Другими словами, реконфигурирование – перепрограммирование аппаратуры [1, 2]. Составляющие систему с изменяемой конфигурацией блоки (аппаратные узлы) могут быть реконфигурированы единожды (в момент выпуска вычислительной системы), с определенной периодичностью, для ликвидации багов, для обновлений и улучшений системы, либо для адаптации под конкретную решаемую задачу в текущий момент времени. Runtime (динамическая) реконфигурация – реконфигурация архитектуры вычислительной системы в процессе её функционирования (рисунок 1), статическая – перед запуском (рисунок 2).

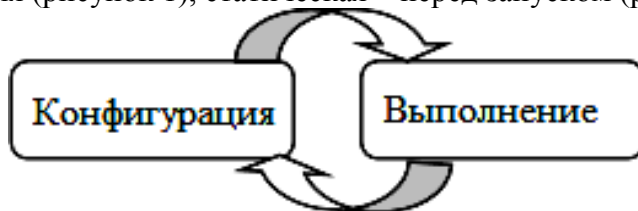


Рисунок 1 – Динамическая реконфигурация

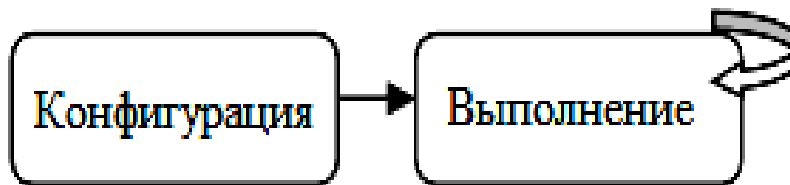


Рисунок 2 – Статическая реконфигурация.

Как правило, любая вычислительная система состоит из тракта обработки данных (datapath) – набора функциональных элементов, образующих собой тракт обработки данных, и контроллера, подающего на вход тракта управляющие последовательности команд, включающие в себя данные для обработки. Как правило, реконфигурируемость касается только тракт данных, контроллер неизменен. Кроме того, реконфигурируемые компоненты различаются степенью гранулярности (мелкогранулярная и крупногранулярная). Мелкогранулярная архитектура представляется как программируемая матрица, состоящая из конфигурируемых в один логический элемент блоков, вентилях, и элементов, осуществляющих соединения между этими блоками. Подобные структуры могут изменяться в значительной степени, обладая большой гибкостью, однако, представляет значительную сложность процесс их программирования. Крупногранулярная архитектура представляется цельными, уже готовыми элементами с реконфигурируемыми соединениями между ними. Подобная технология позволяет эффективнее реализовывать сложные операции на основе блоков тракта обработки данных, предоставляет готовые инструменты для конфигурирования аппаратуры, но более ограничена в применении. Создание РВС является сложной и комплексной задачей, предполагающей под собой набор мер, которые требуются для включения реконфигурируемых элементов в некую вычислительную систему (или создания полностью реконфигурируемых вычислителей). Ключевой проблемой является сложность создания конфигурации аппаратуры, решающей конкретную задачу.

Обзор существующих решений. Обычно реконфигурация реализуется при помощи записи значений в некоторые регистры. Грубо говоря, любую память, применяемую в современных компьютерах, можно назвать реконфигурируемой системой.

Мелкогранулярная реконфигурация реализуется, в основном, с применением ПЛИС. Существует решение для построения мелкогранулярных вычислительных платформ, называемый «вычисления с памятью» – когда все значения некоторых функций хранятся в реконфигурируемой памяти. Соответственно, реконфигурация производится простой перезаписью этих значений, а вычисления – обращениями к ним. Это решение с детальным обоснованием выбора описаны в [3-5]. Крупногранулярных решений множество. В качестве примеров можно привести MorphoSys и ADRES [6,7], представляющие собой системы из блоков с меняющимися связями между ними.

Для организации реконфигурируемости в процессе функционирования необходим некий постоянно функционирующий процессор с жесткой и неизменной архитектурой, который обладает способностью изменять структуру реконфигурируемого блока (для ПЛИС это программатор). Один из подходов [8] предполагает наличие менеджера ресурсов, который перераспределяет вычислительные задачи между различными блоками (CPU, DSP, GPU, FPGA) в зависимости от требуемой для решаемой задачи вычислительной мощности. Методология основывается на использовании расширенного компилятора, который разделяет исходный код приложения на модули, генерирует исполняемый код (или генерирует конфигурацию аппаратуры), и анализирует модули. Система реализуется на OpenCL и состоит из отдельных ядер. Каждое из этих ядер (модулей) компилируется и оптимизируется для различных вычислительных ресурсов (CPU, DSP, GPU, FPGA), после чего производится анализ

его производительности. На выходе компилятора получается информация для менеджера ресурсов и набор модулей с дополнительной информацией, необходимой для управления ими.

Технологический процесс генерации аппаратуры для реконфигурируемых систем во многих случаях практически ничем не отличается от подобного процесса для создания заказной аппаратуры, ASIC, поэтому используемые там подходы могут быть полезными в рассматриваемой области. Кроме того, интерес представляют ВС, позволяющие добиться значительного распараллеливания вычислительного процесса и его оптимизации.

Поэтому, довольно перспективными видятся ВС с динамическим микро- (или нано-) программированием. Вычислительная система с динамическим микропрограммированием подразумевает наличие процессора, управляющего поведением datapath посредством задания всех его входных сигналов напрямую, не прибегая к инструкциям. При таком подходе убирается лишняя при автоматической генерации аппаратуры, машинного кода и симулятора абстракция инструкций, что позволяет значительно упростить процесс. Одной из наиболее проработанных систем, использующих данный подход, является архитектура NISC и прилагающийся к ней инструментарий, основывающийся на языке GNR [9-12, 13-15].

Выводы. Существует множество разных типов реконфигурируемых систем, основных принципов их организации и функционирования. В основном применяются подходы, использующие ПЛИС и подобные подходы. Однако, до сих пор ведутся поиски эффективной методологии создания реконфигурируемых систем, генерации конфигураций для них из высокоуровневых спецификаций, и включения в некоторый базовый вычислительный процесс. Нет некоторого универсального и удобного решения, которое стало бы стандартом де-факто, вроде распространённой для процессоров общего назначения фон-Неймановской архитектуры. И нерешёнными, и активно исследуемыми являются проблемы высокоуровневого проектирования систем с реконфигурируемыми модулями, а также сопутствующий процесс распараллеливания существующих последовательных алгоритмов.

Исследование выполнено за счет гранта Российского научного фонда № 21-71-00110, <https://rscf.ru/project/21-71-00110/>.

1. Сопряжённое проектирование встраиваемых систем (Hardware/Software Co-Design) / С. В. Быковский, Я. Г. Горбачев, А. О. Ключев, А. В. Пенской. – Санкт-Петербург: НИУ ИТМО. – Часть 1 – 2016. – 108 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/91380> (дата обращения: 07.10.2021). – Режим доступа: для авториз. пользователей.

2. Каляев И.А., Левин И.И. / под общей редакцией И.А. Каляева. Южный федеральный университет. – Таганрог: Издательство Южного федерального университета, 2016. – 472 с.

3. Reliability improvement in multicore architectures through computing in embedded memory / H. Hajimiri et al., MWSCAS, 2011.

4. Nanoscale Reconfigurable Computing Using Non-Volatile 2-D STTMRAM Array / S. Paul, S. Chatterjee, S. Mukhopadhyay and S. Bhunia, International Conference on Nanotechnology, 2009.

5. Profiling and Online System-Level Performance and Power Estimation for Dynamically Adaptable Embedded Systems / J. Mu, K. Shankar, R. Lysecky., ACM Transactions on Embedded Computing Systems (TECS), Vol. 12, No. 3, Article 85, pp. 1-20, 2013.

6. MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications / H. Singh, IEEE Transactions on Computers. 2007

7. ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix B. Meietal., Springer LNCS, Field Programmable Logic and Application Lecture Notes in Computer Science Volume 2778, 2003, 61-70
8. Arno Luppold. A new concept for system-level design of runtime reconfigurable real-time systems / Arno Luppold [и др.] / ACM SIGBED Review - Special Issue on the 5th Workshop on Adaptive and Reconfigurable Embedded Systems, 2013 №10 (4) pp. 57-60.
9. ESL Design and Verification. A Prescription for Electronic System Level Methodology / Grant Martin, Brian Bailey, Andrew Piziali., 2007.
10. Processor Description Languages / Prabhat Mishra, Nikil Dutt, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2008.
11. Reconfigurable Computing Using Content Addressable Memory for Improved Performance and Resource Usage / S. Paul and S. Bhunia, Design Automation Conference, 2008.
12. B. Gorjiara and D. Gajski / FPGA-friendly Code Compression for Horizontal Microcoded Custom IPs, in International Symposium on Field-Programmable Gate Arrays (FPGA), 2007.
13. Сайт технологии NISC: <http://www.cecs.uci.edu/~nisc/> (Дата обращения: 01.11.2021).
14. NISC Communication Interface / B. Gorjiara, M. Reshadi, D. Gajski, Center for Embedded Computer Systems (CECS) Technical Report TR 06-05, 2006.
15. A scalable memory-based reconfigurable computing framework for nanoscale crossbar/ S. Paul and S. Bhunia. IEEE Trans. on Nanotechnology, 2010.