ОБЗОР ПРЕИМУЩЕСТВ СОВРЕМЕННЫХ АППАРАТНО–ПРОГРАММНЫХ КОМПЛЕКСОВ В РЕШЕНИИ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

Воркунов А.В.[1], Шанаев О.Т.[2] Email: Vorkunov1133@scientifictext.ru

[1]Воркунов Андрей Владимирович – магистрант;
[2]Шанаев Орынгали Толегенович – кандидат педагогических наук, профессор,
кафедра математического моделирования и программного обеспечения,
Алматинский университет энергетики и связи, г. Алматы, Республика Казахстан

Аннотация: *в статье рассматриваются аппаратно-программные комплексы как средство повышения надежности и быстродействия современных компьютерных систем. Описывается концепция использования выделенных аппаратных средств (сопроцессоров, микроконтроллеров, специализированных устройств) для решения вычислительных и прикладных задач. Анализируется применение модульной архитектуры при разработке описываемых систем. Рассматриваются конкретные примеры реализации аппаратно-программных решений. Приводится обзор сфер применения и преимуществ данных решений.*

Ключевые слова: *аппаратные средства, сопроцессоры, микроконтроллеры, модульная архитектура, отказоустойчивость.*

AN OVERVIEW OF THE ADVANTAGES OF MODERN HARDWARE-SOFTWARE COMPLEXES IN COMPUTING TASKS

Vorkunov A.V.[1], Shanaev U.T.[2]

[1]Vorkunov Andrey Vladimirovich – graduate student;
[2]Shanaev Uryngali Tolegenovich - Candidate of Pedagogic Sciences, Professor,
DEPARTMENT OF MATHEMATICAL MODELING AND SOFTWARE,
ALMATY UNIVERSITY OF POWER ENGINEERING AND TELECOMMUNICATIONS, ALMATY, REPUBLIC OF KAZAKHSTAN

Abstract: *the article considers hardware-software complexes, as a means of improving the reliability and performance of modern computer systems. The concept of using dedicated hardware (coprocessors, microcontrollers, specialized devices) for solving computing and applied objectives is described. The application of the modular architecture in the development of the described systems is analyzed. The article discusses specific examples of implementation of hardware-software solutions. Provides an overview of the application spheres and benefits of these solutions.*

Keywords: *hardware, coprocessors, microcontrollers, modular architecture, fault tolerance.*

УДК 004.272.32

Сложность задач, возлагаемых на компьютерные системы, непрерывно растет, как и требования к их надежности, безопасности и быстродействию. В современных условиях возникает необходимость к новому подходу в разработке ПО для решения вычислительных и прикладных задач. Одним из эффективных способов значительного повышения надежности и быстродействия вычислительных систем является применение выделенных аппаратных средств для решения узкоспециализированных задач.

Сейчас аппаратно-программные комплексы широко распространены в различных отраслях производства: робототехнике, машиностроении и других, где вычислительные функции не являются основными, однако их применение не столь распространено в информационных системах. Тем не менее, эффективность переноса определенных функций ПО на: микроконтроллеры, ПЛИСы и дополнительные микропроцессоры для решения ряда задач на ПК-платформе уже доказана на практике. Преимуществами такого подхода являются повышенная надежность и производительность, а также модульность системы.

Одним из первых таких примеров являются графические процессоры (GPU), разработанные для ускорения обработки графической информации. Благодаря параллельной архитектуре и аппаратной реализации набора команд они намного производительнее ЦП в задачах многопоточной обработки данных. Высокая производительность GPU объясняется особенностями архитектуры. Если современные CPU содержат несколько ядер (на большинстве современных систем от 2 до 128, по состоянию на 2015 г.), графический процессор изначально создавался как многоядерная структура, в которой количество ядер может достигать несколько тысяч [1]. Разница в архитектуре обусловливает и разницу в принципах работы. Каждое ядро CPU представляет собой универсальный блок АЛУ с кэш-памятью и шинной данных, в то время как в GPU одно ядро способно выполнять строго одну операцию, что позволяет многократно сократить время выполнения одного машинного цикла. Таким образом, сложные

вычисления по обработке полигонов, трехмерных примитивов, карт нормалей, шейдеров и текстурированию имеют аппаратную реализацию на соответствующих блоках GPU и выполняются за один машинный цикл, в то время как на CPU их потребовалось бы несколько. С появлением 3D-ускорителей появились и первые программные API (OpenGL, DirectX, Mantle API, VDPAU и др.) для доступа к функциям GPU на уровне операционной системы. По аналогичному принципу работают и аппаратные аудио/видео декодеры (VC-1/WMV3, MPEG-2, MPEG-4, DivX, Xvid, H.264 AVC), которые на ПК всё чаще встраиваются в CPU и GPU, но также встречаются и в виде отдельных микросхем, например на платах видеозахвата или цифровых ТВ-тюнерах. В настоящее время GPU являются неотъемлемой частью широко распространенных 3D-ускорителей и стали частью ПК-платформы. Появление GPU и графических API спровоцировало активное развитие трехмерной графики, которая нашла применение в системах математического и инженерного моделирования, медицине, кинематографе и видеоиграх.

Другим примером являются современные криптографические установки, основанные на ПЛИС, которые работают по схожему принципу. Так как алгоритмы шифрования предполагают выполнение одного и того же набора операций вне зависимости от входных данных, то вполне логично выполнять их на отдельном устройстве. Первым таким устройством совместимым с ПК является проект АНБ США – криптокарта Fortezza стандарта PCMCIA. В ней применялся алгоритм шифрования SkipJack, который выполнялся на специализированном криптографическом RISC-микропроцессоре Capstone [2]. В настоящее в сфере информационной безопасности существует множество пк-совместимых шифровальных устройств с поддержкой алгоритмов (AES-256, RSA, Twofish, RC4 и др.) на основе сетевого оборудования (Cisco ASA, Cisco TrustSec NGFW, NGIPS и др.) [3]. При использовании подобных устройств ПК работает только с крипто-ключами и хэшами для доступа к информации, в то время как вся остальная работа выполняется на устройстве шифрования. Это позволяет не только разгрузить ЦП компьютера и ускорить процесс шифрования, так как аппаратно оно выполняется быстрее, чем программно на ЦП, по вполне очевидным причинам, но и повысить безопасность всей системы. Дело в том, что шифрование и хранение ключей осуществляется в самой плате шифратора, а не в ОЗУ компьютера, что исключает возможность перехвата ключей. Помимо этого аппаратный датчик случайных чисел генерирует действительно случайные числа для формирования более стойких ключей и электронных цифровых подписей [4].

Кроме более высокой производительности и безопасности, еще одним преимуществом внедрения аппаратно-программных решений является модульная архитектура, которая сейчас активно применяется при проектировании самой различной электроники. Чтобы глубоко не вдаваться в детали реализации коммерческих решений, в качестве примера приведу разработанный мной в дипломной работе мобильный цифровой осциллограф на базе микроконтроллера PIC18. Он построен на модульной архитектуре, центральным управляющим элементом которой выступает сам микроконтроллер PIC18F452 и к нему подключается вся внешняя периферия: щупы, кнопки, ЖК-дисплей, USB-порт для подключения к ПК и зарядки аккумулятора. Схема предусматривает распайку и установку отдельной микросхемы АЦП для повышения характеристик осциллографа. Во-первых, при таком подходе, значительно упрощается общая структура и схема разрабатываемого устройства. Во-вторых, в такой системе все модули независимы друг от друга, благодаря чему очень удобно изменять функционал системы, удаляя или добавляя новые модули. Это сильно облегчает отладку и тестирование устройства, что значительно ускоряет его разработку от выпуска прототипа до производства серийного образца. В-третьих, такое устройство легко модифицировать, как на аппаратном, так и на программном уровне. Так как при сохранении интерфейсов, согласно выбранным стандартам, можно изменять аппаратные возможности устройства (например, добавить дополнительные каналы, дискретный АЦП и т.д.), прошивку микроконтроллера и софт на стороне ПК независимо друг от друга. Также подобным примером являются довольно распространенные сейчас аппаратные сторожевые таймеры Watchdog для серверов, многие из которых основаны на микроконтроллерах. Сторожевые таймеры являются одним из элементов базовой периферии микроконтроллеров и используются для периодического перезапуска устройства, что может быть полезно при нарушении его работы. Для этих же целей они начали применяться и в серверных решениях. Такие сторожевые таймеры позволяют автоматизировать процесс перезагрузки и перезапуска серверов или рабочих станций при их зависаниях, которые могут быть вызваны сбоями в работе ПО, хакерскими DDoS–атаками и пр. В противном случае сервер останется недоступным до его ручного перезапуска. Программная часть на стороне сервера реализуется в виде службы, которая работает в фоне и поддерживает связь с подключенным микроконтроллером, который фиксирует сообщения от сервера. Если микроконтроллер перестает получать данные сообщения, следовательно, сервер перестал нормально функционировать и сторожевой таймер посылает сигнал аппаратной перезагрузки (RESET) на материнскую плату сервера.

Последнее значительное преимущество – это повышенная надежность. Очевидно, что в предыдущих примерах, функционал, реализованный на самостоятельных устройствах, будет надежнее, так как его

работа не зависит от ПК, в свою очередь ПК получает больше свободных ресурсов, что в теории делает его работу более стабильной. Похожий принцип используется и в современных серверных платформах для распределения и балансировки нагрузки. Задача распределения ресурсов является одной из важнейших в процессе виртуализации. На сегодняшний день наиболее эффективным способом её решения является использование многопроцессорных конфигураций. Большинство современных гипервизоров (PowerVM, Hyper-V, VMWare ESXi, Xen, KVM, OpenVZ) поддерживают выполнение виртуальной машины в отдельном потоке, ядре или на отдельном процессоре, что позволяет изолировать её работу от всех остальных. То есть неполадки или сбой в работе одной из виртуальных машин никак не отразятся на работе всех остальных и всей системы в целом. Это особенно актуально для хостинг-провайдеров и облачных сервисов, где на одном сервере может быть запущено от десятка до сотни виртуальных машин [5]. Данная концепция становится настолько популярной, что постепенно переносится с корпоративного рынка на потребительский. Ярким примером является последняя игровая консоль PlayStation 4 от Sony, в которой применены сразу 2 процессора: основной AMD Jaguar (на архитектуре x86-64) и дополнительный SCEI CXD90025G (на архитектуре ARM) [6]. На микропроцессоре SCEI запускается всё системное ПО включающее: операционную систему, драйвера, обновления, оболочку и онлайн-сервисы, в то время как основной ЦП от AMD обрабатывает только запущенные игровые приложения. Такая схема позволяет не только более полно использовать все вычислительные ресурсы ЦП, но и по аналогии с серверными гипервизорами изолировать выполнение приоритетных задач от всех остальных.

Исходя из приведенных выше примеров, можно с уверенностью сказать, что применение программно-аппаратных решений является современной тенденцией в IT-индустрии, и их актуальность и востребованность в ближайшем будущем будет только расти.

Список литературы / References

1. *Таненбаум Э., Остин Т.* Архитектура компьютера. 6-е издание. И.: Питер, 2015 г. С. 612 – 623.
2. *Фоменков Г.В.* Академия ФСБ РФ. Криптокарта FORTEZZA -правительственные технологии в коммерческих приложениях. [Электронный ресурс]. Режим доступа: http://citforum.ru/internet/securities/fortezza.shtml/ (дата обращения: 10.01.2017).
3. Доклад с Cisco Expo 2011 «Архитектура Cisco TrustSec. Профилирование сетевых устройств и шифрование трафика». [Электронный ресурс]. Режим доступа: http://www.cisco.com/c/dam/global/ru_ua/assets/expoukraine2011/pdfs/4_architecture_cisco_trustse_profiling_of_network_devices_and_encryption.pdf/ (дата обращения: 07.03.2017).
4. *Пустовой Д.* «Сетевое шифрование: программное или аппаратное?» // Журнал «Information Security / Информационная безопасность», 2014. № 4. 59 с.
5. *Толети Б.П.* Гипервизоры, виртуализация и облако: О гипервизорах, виртуализации систем и о том, как это работает в облачной среде» // IBM Developer works, 2012 г. [Электронный ресурс]. Режим доступа: http://www.ibm.com/developerworks/ru/library/cl-hypervisorcompare/index.html/ (дата обращения: 12.12.2016).
6. Технические характеристики Playstation 4. [Электронный ресурс]. Режим доступа: http://en.rfwiki.org/wiki/PlayStation_4_technical_specifications/ (дата обращения: 17.02.2017).