

Таблиця 5. Значення часу та розрахункові критерії

№ ітерації	Час роботи випромінювача, с	τ_{42} , с	τ_{\max} , с
1	60	475	282
2	90	470	272
3	120	5200	552
4	150	5300	630
5	180	5300	765
6	210	5300	695
7	240	1575	475
8	270	810	340
9	300	897	385
10	330	885	385
11	360	865	370
12	390		

Виходячи з даних (Таб.5), отриманих при ітераціях, на 12 ітерації умова не була виконана. В результаті, в другому випадку для МИТ-11 реакції біологічного об'єкта у вигляді зміни температури ефективно позитивним буде час 6 хв. 10 с. (370 с), що є більшою тривалістю процедури для одного і того ж БО, ніж в першому випадку.

Висновки

Температурний метод контролю ефективності впливу ультразвука на біологічні структури [4] дозволяє:

- підвищити ефективність терапевтичної процедури, а в ряді випадків, стимулювати терапевтичний ефект, запобігти неконтрольованій дії ультразвуку чи його відсутності під час проведення фізіотерапевтичної процедури (без акустичного контакту випромінювача і БО).
- забезпечити ефект ультразвукової терапії шляхом дії на органи і тканини нормованим сигналом, що корегується і контролюється в реальному часі згідно встановленої дози акустичного випромінювання під час процедури.
- порівняти промодельовані розрахунки для різних режимів озвучування під час УЗТ процедури для одного і того ж біологічного об'єкта та визначити, який з режимів має більшу позитивну тривалість ефективного озвучування.

Література. 1. Миллер Е., Хилл К. Применение ультразвука в медицине. Физические основы. – М.: Мир, 1989. – 568 с. 2. Junge D. Nerve and Muscle Excitation (2nd edition ed.). Sunderland, MA: Sinauer Associates.1981. - pp. 33–37. 3. Anastasiia Kyrylova, Nikolaj F. Tereshchenko. Estimation of ultrasound influence on biological tissue. – XIII International PhD Workshop OWD 2011 – Wisla. 4. Заявка на патент України у 201106659 від 27.05.11 Автоматизований багатофункціональний апарат ультразвукової терапії /Тимчик Г.С., Терещенко М.Ф., Кирилова А.В.

СУБСТАЦИОНАЛЬНЫЕ СТРУКТУРЫ В ПРОЦЕССОРНЫХ ЭЛЕМЕНТАХ

К.А. Сорудейкин

Харьковский национальный университет радиоэлектроники

Relevance Research & Development Corp.

Kirill.Sorudeykin@relvecorp.com

We are considering a new approach to design processors according to the properties of theoretical space being investigated. In this architecture we are uniting a traditional approach along with new principles of computational dynamics. Almost all periphery devices will stay unchanged which gives us more flexibility in deployment of our solution. The only means

which we need is innovatory field of tiny-scale devices or elements designed by man. But we believe that eventually will obtain suitable elementary particles to design our processor units.

1. Введение. Традиционные компьютерные (процессорные) архитектуры основаны на жесткой завязке вычислительных элементов, а так же элементов памяти на тех местах (позициях) и в тех ролях, к которым они были закреплены изначально. Можно сказать, что это некая «жесткая» регулярная структура, в которой до известной степени отсутствует приспособляемость и гибкость к различным особенностям решаемых задач. Так складывается, что эта особенность процессорных архитектур является в то же время их ограничением. И мы покажем, в чем, собственно состоит это препятствие, и что нам необходимо сделать (точнее, что возможно сделать) для его преодоления. Но, обо всем по-порядку. Начнем с анализа стоящей перед нами проблемы.

Процессорные устройства сегодняшнего мира являются в достаточной степени инертными. Это накладывает ограничения на все, что мы можем сделать с помощью этих устройств. Так же, процессорные устройства являются последовательными, поскольку они позволяют решать задачи лишь в первую очередь алгоритмически. Т.е. они, безусловно, содержат такие элементы, как упакованные числа, кэширование, конвейеризацию, гиперпоток и многоядерность. Но, тем не менее, все это – лишь симуляция некоторых возможностей, выходящих за рамки линейности, на базе всё тех же линейных архитектур, одной из характерных особенностей которых является необходимость буфера. Точно так же, как в операционных системах применяется квантование времени, здесь, в процессорных элементах, мы видим эмуляцию распределенности с помощью использования дополнительной памяти.

2. Анализ структуры вычислительного процесса. Но есть одна проблема. Память работает медленно. И она ненадежна. В том плане, что временная память теряет все свое содержимое после прекращения подачи электричества. Распределенность по многопоточности/многоядерности так же имеет те же самые ограничения. Мы имеем фактически несколько работающих вместе устройств, но являющихся в принципе независимыми друг от друга линейными автоматами. Поэтому-то и возникают часто сложности с распараллеливанием тех или иных алгоритмов, что данный тип распараллеливания идет вразрез с естественностью формулирования решаемых таким образом задач. И именно это – нарушение законов естественного формулирования – и ведет к проявляющейся сложности, ограничениям, порой которые обойти невозможно, что, соответственно, ведет к необходимости выделения дополнительных ресурсов, чтобы с помощью различных типов и форм буферизации обеспечить «обход» возникающих проблем. Как результат, мы имеем нарушение первичной «композиционной целостности» задач, полное порой изменение их формы, ввиду необходимости адаптации под различные особенности и специфику архитектуры вычислительных систем. Т.е. иными словами, мы получаем формулировки иных задач, которые, зачастую, могут уже не соответствовать задачам, бывшим в оригинале.

Все дело в той же самой регулярной структуре. Поток, как и ядро, являются однотипными структурами. Их форма не создается под необходимость некоторой задачи. Они существуют непосредственно и, фактически, «диктуют правила» о том, как нам необходимо производить вычисления, моделирование и т.п. В то время как сама задача может требовать чего-то иного, более для нее специфического. Что же мешает нам формировать аналоги тех потоков, которые имеются на сегодняшний день, но делать их более специфичными для решаемых задач? Ответ на этот вопрос ясен – необходимость более гибких структур, лежащих в основе процессора.

3. Динамическое формирование задач. Природная параллельность, которую можно скорее назвать одновременностью, имеет иную природу, нежели та жесткая архитектура, которую мы имеем в ядре процессора. Элементарные единицы вычислений, такие как ячейки памяти, буферы, потоки и т.п., являются слишком жесткими и слишком «грубыми», чтобы иметь возможность отразить природную структуру процессов и задач,

работа с которыми производится при помощи этого процессора. Эти элементарные единицы не отражают в достаточной мере структур и отношений, которые мы имеем в решаемых нами задачах, что ведет к необходимости синтезировать еще одну форму задачи, пригодную для процессора, но отличную от первоначальной, что ведет к утрате ее некоторых свойств.

Здесь возникает самая сложная и противоречивая задача. Дело в том, что естественные формы представления вопросов для компьютерных систем заслуживают очень детального изучения. Тот принцип, который принят сегодня – использование искусственных форм представления задач в форме симметричных блоков памяти, матричных структур, является для нас удобным (по причине своей привычности и, может быть, простоты), но, в то же время, является и громоздким. Готовые элементы вычисления в форме стационарных потоков не обладают той гибкостью, чтобы представить структурные единицы задач, полученные в результате их декомпозиции, что препятствует процессорам быть продуктивными в анализе настолько же, насколько и естественный интеллект. Потоки, если есть необходимость сохранять это название, должны выделяться со всеми характеристиками, соответствующими *конкретной задаче*. Но тогда и теряется смысл самих понятий «поток», «процесс» и т.п., так как мы получаем динамику работы процессора в форме взаимодействующих между собой объектов, что имеет сходство с идеей *квантового компьютера*.

Вот мы и приходим к тому, чтобы сформулировать требования к искомой функциональности процессоров для компьютеров нового поколения. *Каждая минимальная единица вычислительного процесса должна нести в себе максимум информации для своего независимого функционирования, и при этом набор этих «знаний» должен стремиться быть минимальным. При этом глобальные средства конфигурирования, подобно потокам и прочим должны быть по-возможности минимизированы*, поскольку они сами по себе ограничивают гибкость процессора, являясь регулярными и негибкими. Итак, каким же образом возможно это осуществить, т.е. реализовать подобную архитектуру?

Как мы знаем, существующие кремниевые структуры являются достаточно жесткими, поскольку формируются на кристаллах кремния (а некоторые современные – на кристаллах гафния, как более «устойчивого» материала, позволяющего уменьшить топологию и увеличить частоты), с использованием оксидов этих веществ в качестве изоляторов для проводников, а сами проводники изготавливаются из серебра, меди, или сплавов, достаточно легкоплавких, но в то же время выдерживающих заданную нагрузку внутри процессора. Полупроводниковые же элементы изготавливаются путем различных технологий, например «ионного внедрения» из нескольких слоёв, в том числе диэлектрики, металл, оксиды, различные примеси и т.п. Таким образом, эти элементы прочно закреплены на своих местах. А теперь проанализируем проблему этой структуры.

Одна из наиболее частых операций, проводимых компьютером – это копирование памяти. На процессоре, как топологически разделенной структуре, данные периодически переносятся по шинам из одной части процессора в другую – из ядра в алгоритмическое устройство, оттуда в регистры и кэш и т.п. Хотя все эти элементы и расположены определенным наиболее удобным способом, но все же они мешают друг другу, «заслоняют» пути передачи информации и усложняют структуру, делая процессор не таким эффективным, каким он мог бы быть. Такие технологии, как Cache Prediction позволяют «на лету» предполагать различные исходы элементарных операций внутри процессора, чтобы затем использовать подготовленные заранее данные, сохраненные в кэше и таким образом повышать эффективность процессора. Эта технология основывается на вероятностном подходе, но достаточно часто «даёт сбой», т.е. если программа написана не определенным образом, позволяющем компьютеру легко и правильно предугадывать переходы, то тогда компьютер начинает это делать неверно, и

вся выгода от этой технологии пропадает. Линейность (плоскостность) расположения (или любая другая «жесткая пространственность»), т.е. определенная упорядоченность во

взаимной организации и взаимном расположении элементов вычислительного процесса, будь-то расположение физических элементов на кристалле, или же расположение элементарных единиц вычислительного процесса в самом этом процессе, все равно сохраняет ту же самую черту, что и всегда – необходимость копирования и переноса данных из одного места в другое с использованием буферов, но без изменения самой, жесткой инфраструктуры процессора. Ограничением выступает необходимость затрат времени и ресурсов (вычислительных, памяти) для переноса этих данных.

Если же структура вычислительного процесса каким-то образом вдруг меняется (ввиду, например, изменения каких-либо ее начальных условий), это ведет к необходимости вследствие этого перестраивать всю структуру ее представления в памяти, доступ к которой так же осуществляется последовательно, по шинам, по отдельным ячейкам памяти, но никак не по значительным группам, представляющим сразу некоторые «куски» описания задач, которые могут быть оставлены без изменения даже при переформулировании задачи. Таким образом мы можем вовсе и не нарушать целостность, перенося сразу целые участки памяти, не выполняя даже при этом операции копирования, и, следовательно, не используя дополнительных ресурсов. Эта операция в большей степени похожа на естественное поведение мыслящей субстанции.

4. наброски архитектуры. Рассмотрим теперь архитектуру, которая могла бы реализовать задуманную нами идею функционирования процессоров. Очевидно, что для выбранной нами динамики подходят *мультиагентные системы*. Они обладают многокомпонентностью, гибкостью, их элементарные частицы просты и демонстрируют свойство *эмерджентного поведения* при работе совместно. Их, в принципе, можно настроить на возможность работать как ячейки памяти. А можно – и как вычислительные элементы. Но как же нам ввести их в микроскопическую структуру процессора? Здесь мы и должны обратить свой взор в сторону нанотехнологий.

Как известно, нанотехнологии направлены на то, чтобы создавать элементарные структуры и устройства из наиболее элементарных и миниатюрных частиц, доступных в нашем мире. Представители направления *нередуцируемой сложности и разумного замысла* обращают внимание на то, что биологические клетки даже нашего организма содержат сложнейший спроектированный инженерный механизм, включая трубки для транспортировки веществ, биологические микроскопические двигатели, включающие все требуемые для этого составные части, и прочие элементы инженерного мира. Специалисты в области нанотехнологий стараются повторить эти эксперименты в лабораторных условиях, чтобы сделать такие технологии доступными для общественности и для нашей бытовой, профессиональной и научной жизни. Они говорят так же и о самовоспроизводимости, но пока эти планы являются лишь задумками, но не стоят в графиках работ реальных лабораторий, как утверждают источники.

Одним из самых сложных прототипов наноробота является «DNA box», созданный в конце 2008 года международной группой под руководством Йоргена Кьемса [1]. Устройство имеет подвижную часть, управляемую с помощью добавления в среду специфических фрагментов ДНК. По мнению Кьемса, устройство может работать как «ДНК-компьютер», т.к. на его базе возможна реализация логических вентилях. Важной особенностью устройства является метод его сборки, так называемый ДНК оригами (англ.), благодаря которому устройство собирается в автоматическом режиме. В 2010 году были впервые продемонстрированы нанороботы на основе ДНК, способные перемещаться в пространстве. Кроме того, например, другим известным специалистом, Алексом Зеттлом, был продемонстрирован рабочий наномотор с размером ротора около 500 нанометров, но конструктивное исполнение мотора затрудняет его использование в нанотехнологиях. Мотор был изготовлен на основе кремниевой подложки и золотых электродов. В качестве оси для ротора исследователи использовали нанотрубку.

Конструктивные элементы таких устройств – молекулы и атомы, размер которых не превышает единиц нанометров. Из них строятся шестеренки, трубки и всевозможные

иные конструктивные части механизмов. Затем, с помощью реакций различного типа, а так же с помощью иных свойств среды могут быть построены *наномоторы* по примеру существующих моторов в природе. Пример такого мотора – кинезин. Этот белок участвует в транспортировке полезных веществ внутри клеток по специальным трубкам. Среди всех его конструктивных свойств и особенностей можно выделить то, что он является малым в размерах – размер «головки» 7.5x4.5x4.5 нм, длина вместе с «хвостом» - порядка 50 нм. Он делает шаг на длину 8 нм, в сутки может продвинуться вплоть до 500 нм. На каждый «шаг» головки при этом затрачивается энергия гидролиза одной молекулы АТФ и вырабатывается энергия бпН (достаточно высокий показатель). Все эти цифры вполне сопоставимы, например, с масштабами 32нм полупроводниковой производственной технологии, используемой компанией Интел при производстве процессоров.

5. Выводы. Для осуществления данной идеи нужно решить проблему «питания» такого типа процессоров, а так же отвод излишней мощности и температуры. Так же необходимо детально изучать их коллективную динамику, как динамику специального пространства с особыми свойствами, что делается, в частности, автором работы в его исследованиях Теории Взаимодействий [2]. Применяя модели, разработанные в рамках этой теории, а так же многих других теорий, связанных с изучением мультиагентного, коллективного поведения, самоорганизации, конкуренции и проч., можно выработать методы и дисциплину управления такими частицами, способы транслирования сигналов между ними, а, соответственно, и достичь управления процессорами, построенными на их основе, чтобы решать различного типа задачи с их помощью. В режиме светового или электронного функционирования скорость таких процессоров будет соответствовать или превосходить существующие на сегодняшний день, а в режиме реконфигурирования они так же будут работать быстро – сравнимо с биологическими компьютерами или естественным интеллектом, а так же микроконтроллерами. Разница между скоростями в первом и втором режиме для человека будет неощутима. Такие процессоры смогут коллективно переносить участки памяти из одного места в другое, решая пространственным образом те задачи, для которых в традиционных архитектурах требовалось бы линейное время.

Литература. 1 Kurt V. Gothelf & Jørgen Kjems at al. Self-assembly of a nanoscale DNA box with a controllable lid. The Nature Journal 459, 73-76 Accepted 6 March 2009.
2 Kirill Sorudeykin. "An Operational Analysis and the Degree of Inertia in Thinking Process Modeling". Proceedings of VI-th international conference "Science and Social Problems of Society: Informatization and Informational Technologies", KhNURE, 2011.