

Д.И. Приходько, А.В. Мокряков, В.В. Горшков

ОБЗОР БАЗОВЫХ МЕХАНИЗМОВ МУЛЬТИРАЗРЯДНОГО МИКРОПРОЦЕССОРА, СОВМЕСТИМОГО С АРХИТЕКТУРОЙ AMD64

Аннотация. Представлены необходимые доработки для улучшения микропроцессоров, которые предназначены для эксплуатации вычислительных систем в различных экстремальных условиях окружающей среды, в частности в космосе. Рассмотрены технические аспекты процесса смены разрядности вычислений (режима работы) микропроцессора под воздействием агрессивной окружающей среды. Этот эффект направлен на повышение надежности работы микропроцессора; с этой целью рассмотрены механизмы микропроцессора, а также сопроцессора как одного из компонентов современных микропроцессоров. Также рассмотрен принцип динамического управления операциями чтения/записи в регистр и новые операции: понижение разрядности вычислений, что подразумевает смену режима работы микропроцессора, и снижения числа ядер – «логическое» изолирование поврежденного ядра в микропроцессоре от работающего функционала. Для сопроцессора предложен механизм раздельного хранения компонент чисел с плавающей точкой стандарта IEEE-754 и показаны способы его реализации.

Ключевые слова: мультиразрядные микропроцессоры, смена разрядности вычислений, реляционная структура регистра.

D.I. Prikhodko, A.V. Mokryakov, V.V. Gorshkov

OVERVIEW OF THE BASIC MECHANISMS OF A MULTI-BIT MICROPROCESSOR COMPATIBLE WITH THE AMD64 ARCHITECTURE

Abstract. The article focuses on the necessary improvements to the microprocessors that are designed to operate computer systems in various extreme environmental conditions, especially space. Technical aspects of the process of changing the bit ratio (operating mode) of the microprocessor under the influence of aggressive environment are viewed. This effect is aimed at improving the reliability of the microprocessor, and for these purposes the mechanisms of the microprocessor as well as the coprocessor (as one of the components of modern microprocessors) are considered. The principle of dynamic control of read/write operations to the register is addressed, as well as new operations were considered - reducing the bit capacity of calculations, which implies changing the mode of operation of the microprocessor, and reducing the number of cores – “logical” isolation of the damaged core in the microprocessor from the operating functionality. A mechanism for separately storing floating point components of the IEEE-754 standard has been proposed for the coprocessor, and methods of implementation have been shown.

Keywords: multi-bit microprocessors, bit change of calculations, relational register structure.

Введение

Сегодня для применения современных микропроцессоров существует большой перечень механизмов, которые значительно улучшают функционирование микропроцессоров [1]. При этом существует множество способов улучшения функциональных возможностей микропроцессоров в разных условиях окружающей среды.

Особо стоит отметить усовершенствование механизмов работы со своей внутренней памятью (регистрами), разрешающее при необходимости перестроить последовательность триггеров в регистре; при этом применяется не физическое расположение тригге-

Приходько Дмитрий Игоревич

выпускник магистратуры, инженер вычислительных и электронно-вычислительных машин, Научно-исследовательский институт системных исследований Российской академии наук, Москва. Сфера научных интересов: информационные системы, вычислительные системы, операционные системы, эксплуатация вычислительных систем. Автор более 60 опубликованных научных работ. SPIN-код: 9063-2855, AuthorID: 968616.

Электронный адрес: mitry1205@mail.ru

Мокряков Алексей Викторович

кандидат физико-математических наук, научный сотрудник, Московский авиационный институт; научный сотрудник, Российский государственный университет имени А.Н. Косыгина «Технологии. Дизайн. Искусство», Москва. Сфера научных интересов: информационные системы, вычислительные системы, математическое моделирование. Автор более 100 опубликованных научных работ. SPIN-код: 8582-8288, AuthorID: 725049.

Электронный адрес: ali.latex@gmail.com

Горшков Владимир Владимирович

доктор технических наук, профессор, заведующий кафедрой прикладной математики и системного программирования, Российский государственный университет имени А.Н. Косыгина «Технологии. Дизайн. Искусство», Москва. Сфера научных интересов: информационные системы, вычислительные системы, операционные системы, эксплуатация вычислительных систем. Автор более 100 опубликованных научных работ. SPIN-код: 9778-6838, AuthorID: 1131369.

Электронный адрес: gorshkov-vv@rguk.ru

ров в регистре, а логическая последовательность хранения номеров триггеров, к которым идет обращение через r/w-операции.

Табличная организация работы регистров в микропроцессоре может использоваться в нескольких случаях:

при необходимости реализации динамической смены системы команд в микропроцессоре. Используется в ситуации совмещения на одном кристалле набора архитектур, потенциально несовместимых в одном режиме работы. Смена идет через перезагрузку компьютера или через особую команду. При этом происходит принципиальная смена адресации;

при необходимости реализации механизма мультиразрядности, которая позволяет снизить вероятность повреждения элементов микропроцессора. Данный механизм применяется в случае, если возникает необходимость в использовании микропроцессора в экстремальных условиях окружающей среды, причем с необходимостью продлить работоспособность микропроцессора ценой потери точности вычислений.

В настоящей статье рассмотрены вопросы, связанные с механизмами мультиразрядности микропроцессора на примере микропроцессора системы команд AMD64 [2], а именно табличная структура регистров в микропроцессоре. Дополнительно рассмотрены улучшения сопроцессора. В качестве примера фактора агрессивной окружающей среды выступает ионизирующее излучение в космическом пространстве. Необходимо отметить, что в статье рассматривается базовая модель без углубления во внутреннюю структуру микропроцессора.

Методология исследования

Рассмотрены основные вопросы, связанные с реализацией табличного механизма в микропроцессоре с целью повышения эффективности смены разрядности вычислений в микропроцессоре (особенно в меньшую сторону); с увеличением разрядности особой проблемы не возникает – точность вычислений лишь возрастает.

В технических изделиях для повышения надежности при работе в экстремальных условиях используют два способа [3].

Дублирование функционала вычислительной системы. Способ подразумевает использование нескольких микропроцессоров (например, три) [4]. При выходе из строя даже двух микропроцессоров система останется в работающем состоянии, хотя и со сниженной работоспособностью.

Отключение неработоспособных компонентов. Под данным способом подразумевается отсечение неработоспособного функционала от решения задач; может быть как логическим, так и техническим. В рамках микропроцессора подразумевается использование операции снижения разрядности вычислений [5].

Под первым способом следует понимать следующее [6]. В рамках задачи по повышению надежности вычислительной системы выбирается мультипроцессорная система, которая содержит несколько микропроцессоров, работающих параллельно и осуществляющих одинаковые вычисления. За счет этого можно добиться существенного повышения надежности вычислений. Тем не менее этот способ не является идеальным для ситуации использования вычислительной системы в экстремальных условиях окружающей среды, хотя он не лишен определенных положительных характеристик.

Основные достоинства данного способа

Простота реализации. Нет необходимости производить внутреннюю доработку микропроцессора. Этот способ используется в технологии HotSwap, для реализации которой внутри системного блока может использоваться специальный контроллер [7], в отказоустойчивых кластерах и других технологиях, которые призваны обеспечить стойкость вычислительной системы к сбоям.

Низкая стоимость компонентов. За счет того, что компоненты задублированы, стоимость такой вычислительной системы в общем случае состоит из суммы стоимостей компонентов.

Однако использование данного подхода становится проблемным при разработке вычислительных систем, которые должны работать в экстремальных условиях окружающей среды.

Под экстремальными условиями окружающей среды следует понимать специфические условия окружающей среды, воздействие которых обладает критическим эффектом, приводящим к нарушениям работоспособности вычислительной системы. Как правило, эти нарушения носят необратимый характер. Такими экстремальными факторами являются: ионизирующее излучение, расплавленная лава, вулканический пепел, цунами и др. Особым случаем из всех перечисленных экстремальных условий окружающей среды является воздействие ионизирующего излучения.

Поэтому, чтобы добиться максимального повышения надежности вычислений, приходится дополнительно использовать второй способ. Метод отключения поврежденной части вычислительной системы основан на том, что вычислительная система имеет избыточный набор функций по отношению к минимально допустимому. При повреждении

части вычислительной системы поврежденный функционал постепенно удаляется из системы физическим (отключается питание) или логическим способом (блокируется адресация на поврежденный элемент системы), и процесс проведения деградации может быть смоделирован на основе результатов работы [8].

Данный способ имеет определенные достоинства и недостатки.

Достоинства:

- позволяет удалить поврежденный элемент системы, что упрощает разработку;
- позволяет разработать систему резервного функционала, который характерен для первого способа (при гибридном подходе).

Недостатки:

- требует разработки серьезных внутренних механизмов;
- стоимость может оказаться выше, чем для системы с резервным копированием.

На практике в вычислительных системах применяются оба способа. В рамках данной статьи рассмотрены вопросы перестройки внутренних механизмов микропроцессора по модели организации стека Фон-Неймана – Лебедева с полным набором команд (CISC) – микропроцессоры Intel последних команд – AMD64.

Прежде чем рассмотреть операцию снижения разрядности вычислений, необходимо отметить, что последние поколения микропроцессоров архитектуры AMD64 имеют очень большой перечень расширенных регистров (см. Рисунок 1).

Поэтому для создания специальных программных средств, в которых будет использоваться механизм смены разрядности вычислений, потребуется использование не самых новых компиляторов. *Это связано с тем, что компиляторы ранних периодов создавали программы, которые должны были работать в крайне ограниченных аппаратных ресурсах вычислительных систем, требующих по этой причине особой осторожности при программировании.*

В связи с этим для последних поколений микропроцессора AMD64 доступны следующие способы изолирования неработоспособных компонентов.

Отключение ядер микропроцессора. Подразумевается отключение ядер и выполнение на каждом оставшемся ядре идентичных вычислений, количество ядер при этом должно быть нечетно; для согласования применяется функция правдоподобия. Производительность системы понижается, точность сохраняется.

Отключение сопроцессора. Способ позволяет высвободить логические адреса триггеров, занятые сопроцессором, что допускает их дальнейшее использование в качестве дополнительных триггеров. Теряется возможность работы с числами IEEE-754, но не теряется точность целочисленных вычислений и возрастает их надежность.

Отключение средств YMM (AVX) и XMM (SSE). Метод исключает расширенные наборы регистров длиной 128 и до 512 бит для целочисленных вычислений и чисел с плавающей точкой. При этом возрастает надежность, однако впоследствии происходит серьезное падение точности вычислений.

Отключение средств MMX. Отключение мультимедийных расширений понижает производительность программ, которые ориентированы на использование мультимедиа, так как набор MMX представляет собой адрес к мантиссе набора x87, но в то же время возрастает надежность вычислений.

Понижение разрядности процессора. Это комплексный способ повысить надежность работы информационной системы, под которым подразумевается откат режима

микропроцессора со старшего режима адресации на младший режим. Для AMD64 таких переходов возможно два – с Long mod на protected mod, и с protected mod на real mod. Переход с 16 бит на 8 невозможен по техническим причинам – i8085 и i8086 имеют двичную несовместимость по адресации.

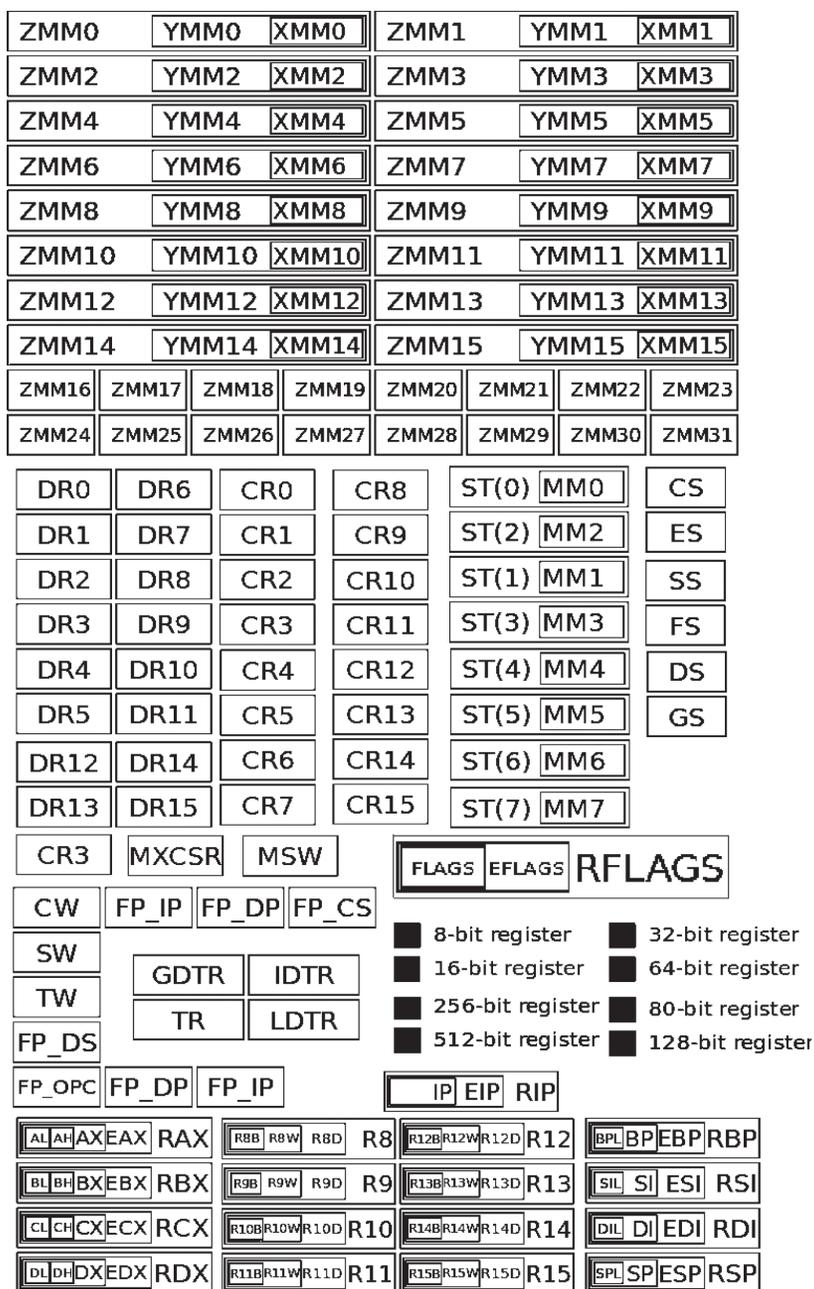


Рисунок 1. Полный набор регистров процессора архитектуры AMD64 с дополнением SSE 4.0
 Источник: [2].

Вышеизложенные методы (за исключением способа понизить разрядность процессора) являются тривиальными и потому не требующими существенных доработок в микропроцессоре [9], поэтому в рамках статьи будет рассмотрен лишь последний способ повышения надежности, в рамках которого реализован табличный механизм работы регистра.

Результаты

Чтобы можно было понизить разрядность вычислений в микропроцессоре, необходимо использовать два типа механизмов:

- механизмы сохранения данных, которые подразумевают методики сохранения целостности данных при изменении режимов работы микропроцессора;
- механизмы сохранения адресации – наименование частей регистра не зависит от физического расположения триггеров в регистре. При смене режима работы микропроцессоров допускается переименование областей регистра.

Механизмы сохранения данных представляют собой использование алгоритмов работы для типов данных, которые присутствовали в программе, и дополнительное резервное копирование в дублирующие триггеры. В простейшем случае их можно классифицировать в нижеперечисленные группы.

Механизм преобразования целочисленных вычислений в вычисления с плавающей точкой основан на том факте, что размеры регистров, которые ориентированы на работу с числами по стандарту IEEE-754 [10], являются сравнительно большими, чем целочисленные регистры. Например, регистры x87 спокойно вмещают в себя числа по типу long double, в то же время набор x86 регистров позволяет в общем случае оперировать числами 2^{16} , что существенно меньше.

Механизм замены типов данных. Под ним подразумевается метод, при котором для представления чисел с плавающей точкой ценой или без потерь в вычислениях используется менее точный тип данных при понижении режима работы микропроцессора. Например, вместо long double используется тип single. Этот подход можно использовать и для представления чисел с фиксированной точкой (они представляют собой смещенное целое число на фиксированное число разрядов). Для целочисленных типов данных операция нехарактерна.

Использование механизмов дублирования триггеров в регистре. Наборы триггеров должны быть занесены в табличную структуру регистров, которая позволяет дублировать хранения чисел и при повреждении триггеров выполнять логическое исключение поврежденных триггеров из вычислительных операций. Этот способ является обязательным дополнением к табличной структуре регистра.

Теперь необходимо рассмотреть **механизмы сохранения целостности адресации**, первостепенной задачей которых является аппаратное обеспечение работоспособности программ при смене режима работы микропроцессора.

К ним можно отнести следующие механизмы:

- табличная структура регистра и реализация операций r/w в регистрах;
- отключение ядер в микропроцессоре и операция понижения разрядности вычислений;
- усовершенствование механизмов сопроцессора для арифметики чисел с плавающей точкой.

Табличная структура регистра и структура выполнения операций с регистром, основанная на табличной структуре, изображены на Рисунках 2 и 3 соответственно.

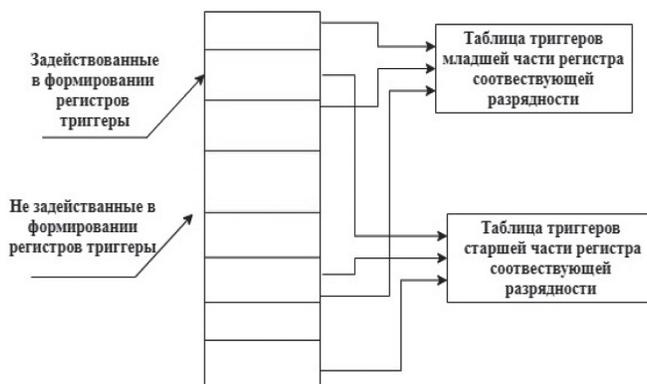


Рисунок 2. Табличная структура регистра в разрезе триггеров, (физическое расположение выбрано случайным образом)

Источник: Рисунки 2–9 составлены авторами.



Рисунок 3. Краткая структура работы табличного регистра для микропроцессора выбранной архитектуры

Приведенные структуры позволяют повысить надежность микропроцессора. Однако для такой структуры регистра необходимо более детально расписать операции r/w. В микропроцессор, который способен быстро перестраивать режим работы в зависимости от накопленных повреждений, необходимо ввести дополнительно две операции – понижения разрядности вычислений и отключения поврежденных ядер микропроцессора.

Структуры выполнения таких операций показаны на Рисунках 4 и 5 соответственно.

На основе анализа Рисунков 4 и 5 можно выделить отдельно *алгоритм обновления структуры триггеров в регистре*. В подробном изложении структуру выполнения алгоритма можно представить в следующем виде.

1. На первом шаге обращаемся к табличному регистру и вытаскиваем данные по составным частям регистра.
2. Собираем «опкоды» обращений к таблице триггеров, включая резервные.
3. Вытаскиваем массив триггеров для чтения значений с учетом анализа резервных копий.
4. На основе пункта 3 формируем двухмерный список с размерностями – триггер и критерии доступа. Критерий доступа имеет тип аналогичный BOOL со значениями: 0 – чтение невозможно и 1 – чтение удачно. При этом проверяем только те итоги, в которых первоначально стоит 1.
5. На основе данных и пункта 4 производим свертку списка – выбираем с критерием доступа 1.
6. На основе пункта 5 создаем табличный регистр.

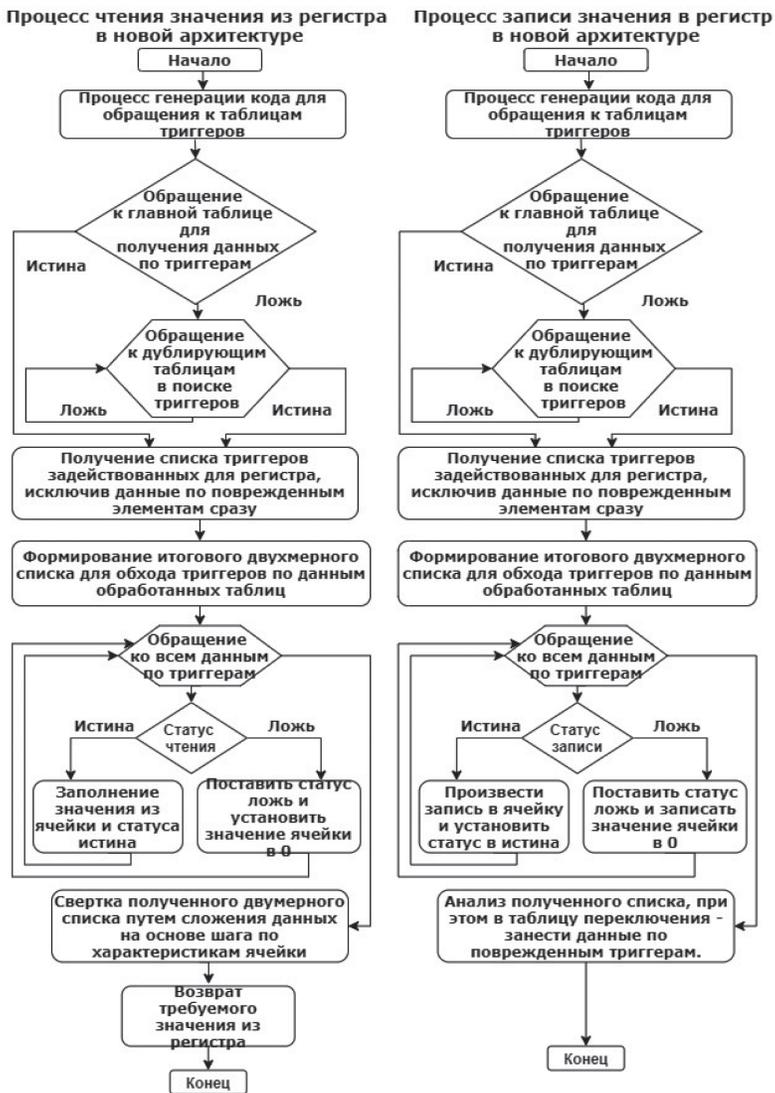


Рисунок 4. Структура операций чтения/записи в регистры микропроцессора на новой архитектуре

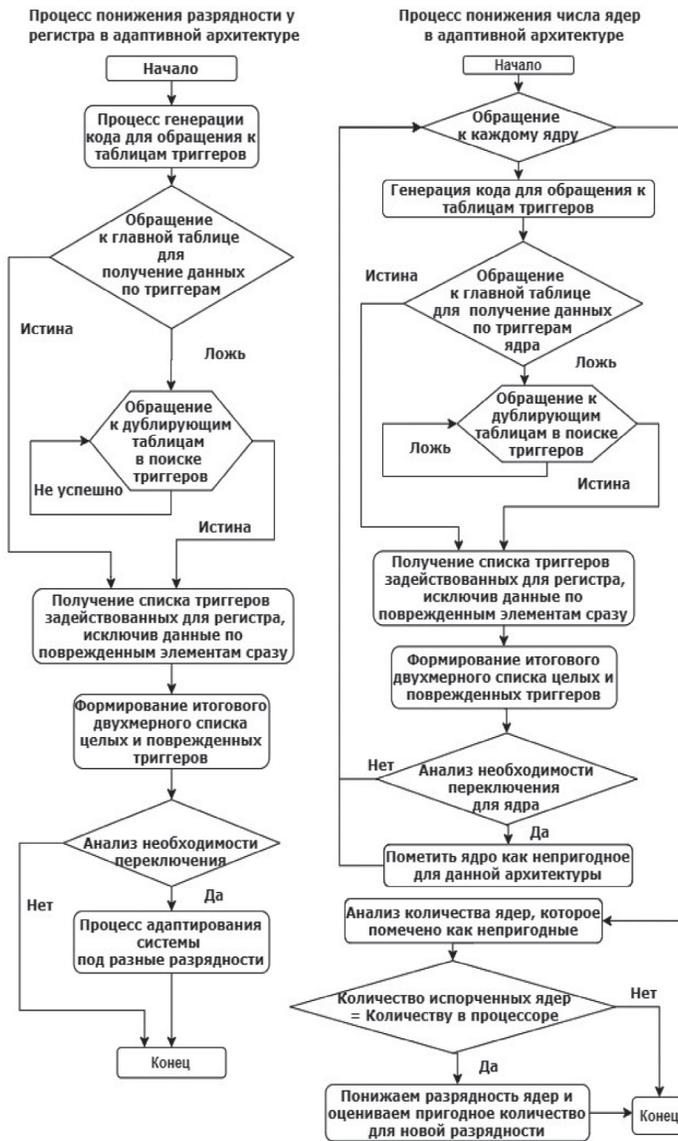


Рисунок 5. Структурная схема операции переключения разрядности регистра и схема сокращения используемых ядер у микропроцессора

После обновления структуры регистра выполнение приведенных в начале статьи операций в регистре можно представить следующим образом.

Операция записи. Выполняется подсчет данных. При совпадении длины числового аргумента и количества ячеек считаем, что операция закончилась успешно, в ином случае результат считается ошибкой.

Операция чтения. Сверяется прочитанное значение с дублерами. Для сверки используются данные шага 2 и прочитанного значения. Выбор осуществляется по принципу преобладающего значения (какое в ячейке представлено в большем количестве копий, то и является достоверным).

Операция понижения разрядности. Определяется кратность набора работоспособных триггеров по отношению к размеру регистра. Если полученный результат меньше допустимого количества резервных копий минус 1 (учитывается сам регистр), выполняется понижение по алгоритму, иначе разрядность менять не требуется.

Операция понижения числа ядер. Анализируется степень повреждения функционала ядер: если есть ядра с повреждениями, при которых работа на текущей разрядности невозможна, ядро изолируется и выполняется перенаправление адресации на другие ядра микропроцессора.

В связи с этим операция понижения разрядности вычислений допустима при соблюдении следующих условий:

- преобразование работы регистров в архитектуре микропроцессора с физического расположения триггеров на реляционную модель базы данных триггеров;
- если микропроцессор состоит из нескольких ядер, то добавляется таблица ядер, а таблица триггеров дополнительно включает в себя измерение – ядро.

Применение описанных механизмов в мультиразрядных микропроцессорах снижает эффективность работы приложений, но повышает надежность их работы.

Описанных механизмов для использования чисел с плавающей точкой недостаточно по следующим причинам:

- низкая устойчивость архитектуры к различным изменениям окружающей среды; при повреждении триггеров в регистре сопроцессора отсутствует возможность однозначно определения местоположения переполнения и повреждений;
- низкая гибкость к внедрению других стандартов чисел с плавающей точкой, отличных от стандарта IEEE-754; примером такого стандарта, который является конкурентом, является Unum-арифметика [11], которая аппаратно не поддерживается в микропроцессоре на момент написания статьи, но, тем не менее, существуют библиотеки эмуляторы данной арифметики для программирования.

Для устранения вышеописанных механизмов предлагается преобразовать регистры набора x87 и более новых, которые представлены в SSE 2.0 и новее, в виде следующей структуры.

Регистр мантисс. Используется для хранения мантиссы из числа с плавающей точкой. Размер регистра выбирается из набора используемых в микропроцессоре float стандартов, например, IEEE-754-2019 или Unum.

Регистр хранения порядка. Используется для хранения порядка из числа с плавающей точкой. Размер регистра выбирается из набора используемых в микропроцессоре float стандартов, например, IEEE-754-2019 или Unum.

Регистр основания. Используется для хранения экспоненты из числа с плавающей точкой. Размер может соответствовать или стандарту IEEE-754-2019 (2 для обычных чисел или 10 для новой редакции), или Unum чисел. Обычно в роли основания применяются основания следующих позиционных систем счисления: 2, 8, 16, 10.

Структура показана на Рисунке 6.

Особенность Рисунка 6 состоит в том, что набор команд MMX, который был ориентирован на использование мантиссы набора x87, аппаратно отделен, что позволяет использовать MMX и IEEE-754 одновременно. Селекторы доступа обеспечивают синхронизацию между частями числа с плавающей точкой. Тем не менее, использование трех регистров, с точки зрения разработчика, менее удобно, нежели одного. Поэтому для упро-

щения разработки программного обеспечения на языках высокого уровня необходимо предусмотреть наборы макросов, которые могут быть выполнены аналогично следующим техническим решениям.

1. Можно дополнить функционал операционной системы специальным интерфейсом, аналогичным DPMI [12]. Используется для приложений Windows 3x и MS-DOS protected mode и поэтому DPMI распространен в семействе Microsoft для 32-битных систем. Windows Me и FreeDOS являются последними системами, обладающими полноценной поддержкой интерфейса DPMI.

2. Также возможно использование аналогичного подхода, который применяется для работы с комплексными числами, – хранение действительной и мнимой частей отдельно. На наш взгляд, можно использовать примеры реализации из языков Matlab, Python, TCL, Perl.

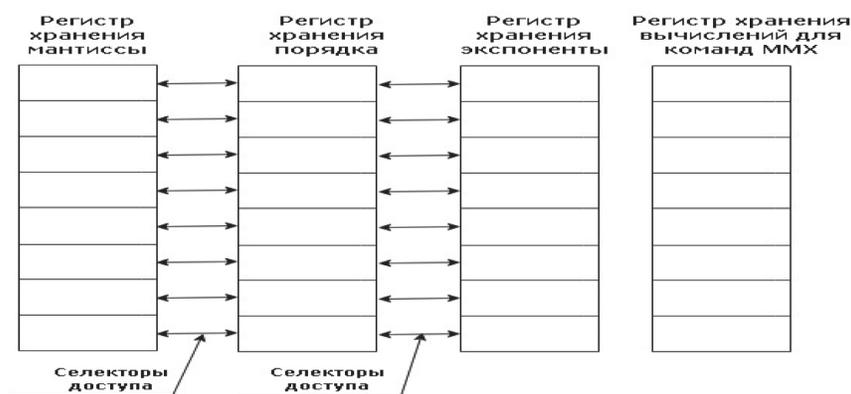


Рисунок 6. Структура регистров сопроцессора на новой архитектуре

Рассмотрим примерный код обращения к двум возможным интерфейсам, аналогичный языку С: для первого способа – на Листинге 1, для второго – на Листинге 2.

```
struct float
{ dword MANTISSA;
  dbyte EXP;
  dword POW;
} float,*float;
float_func_sum( flp_3,flp_4);
```

Листинг 1. Пример реализации структуры DPMI-решения

Применительно к коду, изображенному на Листинге 1, необходимо добавить следующие дополнительные комментарии:

- struct float – структура, сохраняющая в себе отдельные компоненты числа типа float, MANTISSA – мантисса, EXP – основание степени, POW – степень;
- flp_3, flp_4 – числа с плавающей точкой;
- float_func_sum – интерфейсная функция для выполнения операции сложения числа с плавающей точкой на языке С – преобразует структуру в набор аргументов для каждого регистра.

Для другого способа выполнения код иллюстрирован на Листинге 2.

```
//Комплексный метод выполнения сложения:
flp_3=a1*M+b1*EXP+c1*POW;
```

```

flp_4=a2*M+b2*EXP+c2*POW;
//Итог сложения получается по правилам комплексной арифметики:
ITOG=flp_3+flp_4=(a1+a2)*M+(b1+b2)*EXP+(c1+c2)*POW;

```

Листинг 2. Пример реализации структуры DPMI-решения

Применительно к коду, изображенному на Листинге 2, необходимо добавить следующие дополнительные комментарии:

- M,EXP,POW – единичные векторы для измерения мантиссы, основания и степени;
- a1,b1,c1, a2,b2,c2 – составные части чисел flp_3,flp_4 (числа с плавающей точкой),

ITOG – итоговый результат.

Возможно использование как одного из подходов, так и обоих одновременно, причем вариант DPMI, вероятно, будет характерен для программирования драйверов, ОС – системных приложений. Второй подход больше характерен для языков моделирования.

Обсуждение

Как было описано в разделе «Методология», работа отличается в первую очередь тем, что повышение надежности вычислений связано с использованием внутренних механизмов микропроцессора. Необходимо отметить, что возможно использование нескольких способов повышения надежности вычислений [13], которые применяются в технике.

Место работы в научном познании

Практически все рассмотренные при выполнении исследования работы можно разделить на три класса.

Класс работ, направленных на изучение аппаратной устойчивости микропроцессора. Это класс работ, в которых исследователи изучали непосредственное воздействие ионизирующего излучения или других неблагоприятных воздействий на микропроцессор. При этом отдельно рассматривался вопрос, связанный с необходимостью анализа воздействия составных элементов микропроцессора или любой другой микросхемы. Однако в данных работах не рассматривались вопросы, связанные с повышением устойчивости вычислительной системы с точки зрения архитектуры вычислений.

Класс работ, связанных с повышением надежности вычислительных систем в контексте анализа имеющихся ресурсов микропроцессора с точки зрения способа дублирования компонентов вычислительной системы. Первичное рассмотрение механизма было выполнено одним из авторов в работе [9].

Логические работы, посвященные анализу устойчивости элементной базы микросхем, в частности диссертация П.Б. Лагова [8], в которой описывается методика прогнозирования поведения с точки зрения анализа нескольких элементов вычислительной системы. Предложенный способ отличается тем, что опирается не только на внутренние механизмы микропроцессора, но и в большей степени на логическую перестройку архитектуры [14], которая позволяет путем «логического» отсека поврежденных компонентов микропроцессора добиться сохранения работоспособности вычислительной системы ценой потерь в точности вычислений. Поэтому данные особенности нужны исключительно в ситуации, когда вычислительная система используется в экстремальных условиях окружающей среды.

Технический анализ работы

Вышеизложенные механизмы крайне интересны, однако возникает вопрос, каким образом будет происходить воздействие излучением на новую и старую архитектуры ми-

кропроцессора. При этом наиболее существенным фактором является сохранение целостности адресации и числовой точности для конкретного режима работы микропроцессора.

Ответом на этот вопрос с учетом точности вычислений будет сравнение процессов поражения кристалла микропроцессора на двух архитектурах:

- попадание ионизирующего излучения на классическую архитектуру (см. Рисунок 7) – исходная архитектура без дополнительных механизмов;
- попадание на обновленную архитектуру, содержит механизмы из предыдущего регистра.

При попадании на классическую архитектуру микропроцессора происходит поражение триггеров, входящих в регистр (рассматривается ситуация необратимого воздействия излучением), которые постепенно выходят из строя (см. Рисунок 7).

У процесса, изображенного на Рисунке 7, результаты повреждений можно отнести к следующим категориям.

Идеальный сценарий. Означает, что полностью сгорает старшая часть разрядов. Это серьезная потеря точности. Тем не менее, с учетом структуры поражения регистра (см. Рисунок 7), сохраняется адресация для младшей разрядности вычислений. Это очень хороший результат, так как микропроцессор по факту может эксплуатироваться еще какое-то время за счет использования младшей разрядности вычислений.

Негативный сценарий. Означает, что сгорает полностью младшая часть регистра. Точность сохраняется, поскольку самые важные разряды – это старшие результаты, однако старшие разряды не обладают отдельной адресацией для режима IA-32 и AMD64, значит, работать программы на микропроцессоре на младшей разрядности не смогут.

Реальный сценарий. Означает, что регистр будет поврежден крайне неравномерно (затрагивается и старшая, и младшая части), и предсказать, насколько упадет точность при понижении разрядности, то есть удаления старшей части регистра из вычислительных операций, невозможно.

При таких ошибках в лучших случаях можно получить достаточно серьезные потери точности, в худшем – ошибка деления на 0 и выход из строя операционной системы.



Рисунок 7. Логическая структура регистра RAX под воздействием ионизирующего излучения

Таким образом, исходная архитектура (наборы команд x86, IA32, AMD64) является непригодной для нейтрализации последствий воздействия такого излучения, так как излучение попадает случайным образом (проигрывается сценарий 3 при повреждениях).

При использовании доработанной архитектуры получается ситуация, при которой регистр не физический, а «логический», и неважно, куда именно попало излучение, – на старшую или младшую часть регистра, так как за счет переборки можно создать регистр младшей разрядности. Вопрос сохранения «точности» больших чисел, которые не входят в младшую часть регистра, решается численными методами. Одним из способов является, например, перевод целочисленных вычислений в вычисления с плавающей точкой.

С вычислениями для чисел с плавающей точкой можно получить результат, когда при воздействии на регистры старой архитектуры повреждается как мантисса, так и порядок (повреждение идет аналогично изображенному на Рисунке 7 регистру). *Так как точность чисел с плавающей точкой более низкая, чем для целых чисел (по причине использования особенностей округления чисел в стандарте IEEE754, а также по той причине, что можно хранить как большие числа, так и сравнительно малые)*, то точность падает быстрее и сильнее при таком процессе. Однако при использовании дополнительных триггеров в качестве резервной копии можно скорректировать процесс падения точности вычислений. Более того, этот подход позволяет в большинстве случаев сохранить порядок вычислений.

При втором подходе, при котором все данные хранятся отдельно, места повреждений мантиссы и порядка являются очевидными, требуется следующее:

- организация аппаратного способа системного копирования в резервные триггеры данных каждого регистра;
- аппаратные механизмы конвертации типов данных с приемлемой точностью вычислений.

Данные второго подхода позволяют полностью закрыть вопрос о необходимости доработки сопроцессора. При этом важно отметить, что требуемые механизмы будут чрезвычайно эффективны при наличии специального программного обеспечения, которое построено с учетом технических особенностей, изложенных в настоящей статье.

Заключение

Показан один из возможных способов повышения надежности вычислений в микропроцессоре AMD64 с точки зрения анализа воздействия излучением на логическую структуру микропроцессора. При этом рассмотренный способ основан на доработке логической изоляции поврежденной части микропроцессора, то есть относится к типу методов, которые выполняют отключение поврежденного функционала от задач системы.

Проанализированная табличная структура регистра микропроцессора обеспечивает повышение надежности микропроцессора с логической точки рассмотрения, так как позволяет улучшить сохранность и работоспособность микропроцессора по методу отключения неработоспособных компонентов из микропроцессора. При этом при операции r/w в регистре за счет непрерывного контроля целостности и наличия резервных триггеров обеспечивается сохранение точности вычислений на уровне, который доступен приложениям на каждом режиме микропроцессора AMD64. При анализе понижения числа ядер отдельное внимание уделяется использованию внутреннего механизма резервного копирования.

Литература

1. Зыков А.Г., Поляков В.И. Арифметические основы ЭВМ: учебное пособие по дисциплине «Дискретная математика». СПб. : Университет ИТМО, 2016. 140 с. URL: <https://books.ifmo.ru/file/pdf/2078.pdf> (дата обращения 31.10.2022).
2. AMD64 Architecture Programmer's Manual, Volume 3: General-Purpose and System Instructions. Advanced Micro Devices Inc., 2019. URL: https://archive.org/details/advancedmicrodevices_24594_3.28/page/n1/mode/2up (дата обращения 25.0.2021).
3. Артюхова М.А., Кулыгин В.Н. Разработка автоматизированной системы обеспечения радиационной стойкости бортовой аппаратуры космических аппаратов // Инновации на основе информационных и коммуникационных технологий. 2011. № 1. С. 323–324. EDN PEVDUV.
4. Горшков В.В., Приходько Д.И., Мокряков А.В. Устройство для управления конфигурацией вычислительной системы. (2021). Патент на полезную модель 207176 U1, 15.10.2021. Заявка № 2021118609 от 25.06.2021. EDN JBFXGI.
5. Горшков В.В., Приходько Д.И., Мокряков А.В. Устройство для управления разрядностью вычислений. (2022). Патент на полезную модель 209758 U1, 22.03.2022. Заявка № 2021118607 от 25.06.2021. EDN MDXKXC.
6. Полесский С., Жаднов В., Артюхова М., Прохоров В. Обеспечение радиационной стойкости аппаратуры космических аппаратов при проектировании // Компоненты и технологии. 2010. № 9 (110). С. 93–98. EDN MWMEXL.
7. Козлов А., Королев Г. LTC4242 – контроллер «горячего подключения» (Hot Swap) Linear Technologies для управления питанием двух слотов шины PCI Express // Компоненты и технологии. 2007. № 2 (67). С. 144–147. EDN MTFPHL.
8. Лагов П.Б. Повышение импульсно-частотных, тепловых и инжекционных характеристик биполярных кремниевых структур методом радиационно-термической обработки : дис. ... д-ра техн. наук : 05.27.01. М., 2017. 342 с. URL: <https://mpei.ru/diss/Lists/FilesDissertations/303-Диссертация.pdf> (дата обращения 08.06.2021).
9. Приходько Д.И. Разработка и оптимизация методов эксплуатации информационных систем как этапа их жизненного цикла с учетом воздействия агрессивной внешней среды : дис. магистерская : 01.04.02 / РГУ им. А.Н. Косыгина. М., 2021. 204 с.
10. IEEE Standard for Binary Floating-Point Arithmetic. Copyright 2019 by The Institute of Electrical and Electronics Engineers, Inc. URL: <https://standards.ieee.org/ieee/754/6210/> (дата обращения 08.06.2021).
11. Posit-арифметика: победа над floating point на его собственном поле. Часть 1. (Пер. с англ., оригинал: Gustafson J.L., Yonemoto I. Beating Floating Point at its Own Game: Posit Arithmetic). URL: <https://habr.com/ru/post/465723/> (дата обращения 31.10.2022). <http://www.johngustafson.net/pdfs/BeatingFloatingPoint.pdf>
12. Thomason J.G., Horton E.R. US5369770A – Standardized protected-mode interrupt manager. Microsoft Technology Licensing LLC, 1992. URL: <https://patents.google.com/patent/US5369770A/en> (дата обращения 31.10.2022).
13. Чернышов В.Н., Чернышов А.В. Теория систем и системный анализ : учеб. пособие. Тамбов : Изд-во Тамб. гос. техн. ун-та, 2008. 96 с. ISBN 978-5-8265-0766-7.
14. Чистов Д.В., Мельников П.П., Золотарюк А.В., Ничепорук Н.Б. Проектирование информационных систем : Учебник и практикум для СПО. М. : Юрайт, 2019. 258 с. ISBN 978-5-534-03173-7.

References

1. Zykov A.G., Polyakov V.I. (2016) *Arifmeticheskie osnovy EVM* [Arithmetic fundamentals of computers] : Textbook on the discipline «Discrete mathematics». St. Petersburg : University of ITMO Publ. 140 p. URL: <https://books.ifmo.ru/file/pdf/2078.pdf> (accessed 31.10.2022). (In Russian).
2. *AMD64 Architecture Programmer's Manual, Volume 3: General-Purpose and System Instructions*. Advanced Micro Devices Inc., 2019. URL: https://archive.org/details/advancedmicrodevices_24594_3.28/page/n1/mode/2up (accessed 25.0.2021).
3. Artyukhova M.A., Kulygin V.N. (2011) Development of an automated system to ensure radiation resistance of on-board spacecraft equipment. *Innovatsii na osnove informatsionnykh i kommunikatsionnykh tekhnologiy* [Innovations based on information and communication technologies]. No. 1. Pp. 323–324. (In Russian).
4. Gorshkov V.V., Prikhodko D.I., Mokryakov A.V. (2021) Device for controlling the configuration of the computer system. Patent for utility model 207176 U1, 15.10.2021. Application 2021118609 dated 25.06.2021. (In Russian).
5. Gorshkov V.V., Prikhodko D.I., Mokryakov A.V. (2022) Device for controlling the discharge of calculations. Patent for utility model 209758 U1, 22.03.2022. Application 2021118607 dated 25.06.2021. (In Russian).
6. Poleskiy S., Zhadnov V., Artyukhova M., Prokhorov V. (2010) Ensuring radiation resistance of spacecraft equipment in designing. *Komponenty i tekhnologii* [Components and technologies]. No. 9 (110). Pp. 93–98. (In Russian).
7. Kozlov A., Korolev G. (2007) LTC4242 – Controller “hot connection” (Hot Swap) Linear Technologies for power management of two slots bus PCI Express. *Komponenty i tekhnologii* [Components and technologies]. No. 2 (67). Pp. 144–147. (In Russian).
8. Lagov P.B. (2017) *Povyshenie impul'sno-chastotnykh, teplovykh i inzhektionsnykh kharakteristik bipolyarnykh kremnievykh struktur metodom radiatsionno-termicheskoy obrabotki* [Increase of pulse-frequency, thermal and injection characteristics of bipolar silicon structures by radiation-thermal treatment] : Dr. Sci. Diss. Thesis (Engineering Sciences) : 05.27.01. Moscow. 342 p. URL: <https://mpei.ru/dissdiss/Lists/FilesDissertations/303-Dirstation.pdf>. (accessed 08.06.2021). (In Russian).
9. Prikhodko D.I. (2021) *Razrabotka i optimizatsiya metodov ekspluatatsii informatsionnykh sistem kak etapa ikh zhiznennogo tsikla s uchetom vozdeystviya agressivnoy vneshney sredy* [Development and optimization of methods of operation of information systems as a stage of their life cycle taking into account the impact of aggressive environment] : Master's degree diss. : 01.04.02. Moscow. 204 p. (In Russian).
10. IEEE Standard for Binary Floating-Point Arithmetic. Copyright 2019 by The Institute of Electrical and Electronics Engineers, Inc. URL: <https://standards.ieee.org/ieee/754/6210/> (accessed 08.06.2021).
11. Gustafson J.L., Yonemoto I. (2017) Beating Floating Point at its Own Game: Posit Arithmetic. *Supercomputing Frontiers and Innovations*. Vol. 4. No. 2. Pp. 71–86. DOI: 10.14529/jsfi170206
12. Thomason J.G., Horton E.R. (1992) US5369770A – Standardized protected-mode interrupt manager. Microsoft Technology Licensing LLC. URL: <https://patents.google.com/patent/US5369770A/en> (accessed 31.10.2022).
13. Chernyshov V.N., Chernyshov A.V. (2008) *Teoriya sistem i sistemnyy analiz* [System theory and system analysis] : Textbook. Tambov : Tambov State Technical Univ. Publ. 96 p. ISBN 978-5-8265-0766-7. (In Russian).
14. Chistov D.V., Melnikov P.P., Zolotaryuk A.V., Nichepuruk N.B. (2019) *Proektirovanie informatsionnykh sistem* [Designing information systems] : Tutorial and workshop for secondary professional education. Moscow : Yurait Publ. 258 p. ISBN 978-5-534-03173-7. (In Russian).