

ВИКИПЕДИЯ

K8

Материал из Википедии — свободной энциклопедии

K8 — x86-совместимая микроархитектура центрального процессора, разработанная корпорацией AMD. Впервые представлена 22 апреля 2003 года: были выпущены первые процессоры Opteron, предназначенные для серверного рынка. На основе этой микроархитектуры выпускались семейства микропроцессоров Opteron, Athlon 64, Athlon 64 X2, Turion 64. Является кардинально переработанным, значительно улучшенным и расширенным вариантом микроархитектуры предыдущего поколения AMD K7. В новых процессорах удалось преодолеть ряд проблем, являвшихся ахиллесовой пятой K7, а также внесён ряд принципиально новых решений.

Содержание

Основные тезисы

Конвейер K8

Выборка и декодирование инструкций

Исполнение и отставка

Ссылки

Основные тезисы

Микропроцессоры K8 являются суперскалярными, мультиконвейерными процессорами с предсказанием ветвлений и спекулятивным исполнением. Как и процессоры AMD K7 и Intel P6, они теоретически способны исполнять до 3 инструкций за один такт. Как и любой современный x86-процессор, K8 вначале перекодирует внешний сложный CISC набор x86 инструкций во внутренние RISC-

AMD K8

Центральный процессор

Производство	с <u>2003</u> по <u>2014</u>
Производитель	<u>AMD</u>
Частота ЦП	1,6—3,2 <u>ГГц</u>
Частота FSB	800—1000 <u>МГц</u>
Технология производства	130—65 <u>нм</u>
Наборы инструкций	<u>AMD64 (x86-64)</u>
Разъём	
Ядра	<u>Sempron</u>
	<u>Mobile Sempron</u>
	<u>Opteron</u>
	<u>Athlon 64/FX/X2</u>
	<u>Mobile Athlon 64</u>
	<u>Turion 64/X2/X2</u>
	<u>Ultra</u>
← <u>AMD K7</u>	<u>AMD K10</u> →

подобные микрооперации, которые, в свою очередь, уже идут на исполнение. Для повышения производительности в рамках микроархитектуры реализовано спекулятивное исполнение с предсказанием ветвлений и запуском микроопераций Out-of-Order; для снижения влияния зависимостей по данным используются техники переименования регистров, Result forwarding и ряд других.

Конвейер K8

В микроархитектуре K8 используется конвейер с 12 стадиями, значительная часть которых приходится на декодер инструкций.

Выборка и декодирование инструкций

Основная проблема декодирования инструкций x86 заключается в том, что они имеют различную длину (от 1 до 15 байт). В K8 эта проблема решается путём разделения процесса разметки потока инструкций и собственно декодирования на две отдельные подзадачи исполняемые в разных блоках процессора. Дело в том, что перед помещением в кэш первого уровня для инструкций (L1I) проходят через процедуру предекодирования в темпе по 4 байт инструкций за такт. А информация о разметке помещается в специальный массив тегов ассоциированный с *L1*. Благодаря этому упрощается дальнейшее, рабочее декодирование и сокращается конвейер. Такое решение является уникальным, так как в иных процессорах x86 (за исключением K7) используются иные техники решения этой проблемы. Так, в процессорах *Intel P6* разметка осуществляется «на лету», а в *Intel NetBurst* инструкции проходят декодирование **до** сохранения в *L1* (вместо стандартного кэша инструкций используется специальная, довольно сложная структура, хранящая уже декодированные микрооперации — кэш трасс).

K8 располагает двухканальным наборно-ассоциативным кэшем полезным объёмом 64 Кбайт со строкой равной 64 байт. Однако, помимо собственно инструкций, в процессоре также хранится массив тегов разметки — 3 бит на один байт *L1*, то есть порядка 21 Кбайт, а также дескрипторы предсказателя ветвлений — порядка 8 Кбайт.

Из *L1* инструкции выбираются сразу 16-байтными блоками, которые отправляются одновременно, через специальный буфер (*fetch-буфер*), на исполнительный конвейер и в блок предсказания ветвлений (*branch predictor*). В предсказателе ветвлений блок инструкций анализируется с использованием специального буфера адресов переходов (*BTB*) размером 2048 записей и ассоциированных с ним таблиц истории переходов (*BHT*) полной ёмкостью 16К записей, а также некоторых вспомогательных устройств. В случае если в блоке инструкции содержался переход, то выборка следующего блока будет выполнена уже с предсказанного адреса. К сожалению, *branch predictor* — слишком сложное устройство, чтобы работать в полном темпе процессора, поэтому все предсказания выполняются с латентностью 2 такта, то есть, если процессор встречает переход, то следующая выборка из *L1* будет выполнена только через такт. В большинстве случаев эта задержка нивелируется тем, что в одном 16-байтном блоке содержится много инструкций, и общий темп выборки является опережающим.

Из *fetch-буфера* инструкции попадают на декодер. Каждую инструкцию x86 K8 относит к одному из трёх классов:

- *DirectPath* — инструкции, перекодирующиеся в один МОП;
- *DirectPathDouble* — инструкции, перекодирующиеся в 2 МОПа; и
- *VectorPath* — инструкции, декодирующиеся в набор более чем из 2 МОПов.

DirectPath и *DirectPathDouble* считаются простыми, а *VectorPath* — сложными. Фактически в K8 имеется 2 различных блока декодеров, работающие параллельно и дополняющие друг друга. Основной блок представляет собой комплекс из трёх простых декодеров, работающих совместно и декодирующих до трёх *DirectPath*- и *DirectPathDouble*-инструкций за такт в любой комбинации. Второй блок занимается исключительно *VectorPath*-инструкциями и декодирует по одной такой инструкции за такт. Когда работает *VectorPath*-декодер, соответствующие стадии простых декодеров блокируются. Таким образом, декодер K8 можно считать довольно эффективным и производительным устройством, умеющим перекодировать до трёх простых или одной сложной инструкции за такт. В результате декодирования МОПы через промежуточные буферы перепаковываются в специальные группы по три МОПа в группе (лайны). МОПы в группе следуют строго в порядке изначального кода программы, перестановка не производится. МОПы *DirectPath*- и *DirectPathDouble*-инструкций могут смешиваться как угодно (кроме инструкции умножения, которая декодируется в 2 МОПа и всегда помещается в один лайн), МОПы одной *DirectPathDouble*-инструкции даже могут быть расположены в разных лайнах, но все МОПы *VectorPath*-инструкций должны следовать в целом числе групп и не могут смешиваться с МОПами от простых инструкций, что ведёт к некоторой фрагментации и неполному заполнению групп, но, впрочем, не является частой ситуацией, так как абсолютное большинство инструкций в K8 являются простыми.

Исполнение и отставка

Интересной особенностью K8 является то, что процессор внутри себя оперирует целыми группами по 3 МОПа, что позволяет значительно уменьшить количество управляющей логики процессора. В процессорах Intel, хотя МОПы и идут на некоторых стадиях конвейера группами, но каждый МОП всё равно отслеживается отдельно. Ещё одним большим отличием K8 от процессоров *Intel* является то, что он отходит от принципа максимального упрощения микроопераций. Дело в том, что система команд x86 *CISC* содержит большое количество инструкций типа *Load-Op* (загрузка+исполнение) и *Load-Op-Store* (загрузка+исполнение+выгрузка). Так как все современные x86-процессоры являются *RISC*, то такие инструкции внутри процессора разбиваются на большое количество МОПов, каждый из которых выполняет некоторое своё простое действие. Так, инструкция типа `add eax, mem;` будет разложена, по крайней мере, на 2 МОПа — загрузку из памяти и собственно сложения, то есть количество МОПов, которые необходимо исполнить, может значительно превысить количество исходных x86-инструкций, они заполняют внутренние тракты и буферы процессора, не позволяя добиться скорости в 3 операции за такт.

В процессорах микроархитектуры K7 и K8 разработчики решили обойти эту проблему, сделав МОП двухкомпонентным. Каждый МОП в этих процессорах состоит из двух элементарных инструкций: одна микроинструкция целочисленной либо плавающей арифметики + одна микроинструкция адресной арифметики. Таким образом, инструкции типа *Load-Op* и *Load-Op-Store* могут декодироваться в K8 всего в один МОП, что экономит ресурсы процессора и соответственно позволяет повысить эффективность его работы.

При необходимости один из компонентов МОПа может не использоваться и будет заполнен пустышкой. Так, инструкция типа *Load* будет перекодирована только в один МОП, содержащий только адресную компоненту. Нужно сказать, что в новых процессорах *Intel* для ряда инструкций типа *Load-Op* также применён подобный механизм слияния микроопераций в один МОП с последующим его разделением перед запуском МОПа на исполнение, который называется *microfusion*.

Группа из трёх двухкомпонентных МОПов выходит из декодера и дальше контролируется процессором как единое целое с помощью специального блока — *ISU*. Группы МОПов проходят стадии переименования регистров и выделения ресурсов, затем размещаются в *ROB*. В *ROB* группы инструкций хранятся до момента отставки, отставка инструкций производится сразу всей группой, когда в группе будут исполнены все МОПы, и только в порядке очерёдности, заданной исходной программой. Ёмкость *ROB* в *K8* равна 24 группы, что соответствует 72 МОПам либо 144 микрооперациям. В процессоре *K8* используется статическая схема распределения инструкций по исполнительным блокам, то есть, в какую группу *ФУ*^[*неизвестный термин*] будет запущен МОП, напрямую зависит от положения этого МОПа в группе. Всего в процессоре имеется три планировщика инструкций целочисленной и адресной арифметики по числу МОПов в группе.

Из *ROB*-инструкции копируются в буферы планировщиков. В процессоре имеется три очереди планировщиков для *Int*-операций и три для адресных операций, каждая ёмкостью 8 микроопераций. В общем случае из каждой очереди инструкции могут быть запущены на исполнение в *ФУ* независимо друг от друга и с использованием *Out-Of-Order*. То есть инструкции направляются в *ФУ* в том порядке, в котором необходимо процессору. Процессор содержит три 64-битных АЛУ и три *AGU*, подключённых попарно каждый к своему планировщику.

Планировка и исполнение МОПов плавающей арифметики производится в специальном отдельном устройстве. Для их исполнения процессор содержит одно устройство *FMUL*, одно *FADD* и одно *FMISC*, являющееся вспомогательным.

Ссылки

- Исследование архитектуры AMD64 (<http://www.ixbt.com/cpu/amd-hammer-family2.shtml>)
- Семейство процессоров AMD Athlon™ 64 (https://web.archive.org/web/20100610063806/http://www.amd.com/ru-ru/Processors/ProductInformation/0,,30_118_9484,00.html)

Источник — <https://ru.wikipedia.org/w/index.php?title=K8&oldid=126616694>

Эта страница в последний раз была отредактирована 12 ноября 2022 в 18:37.

Текст доступен по лицензии Creative Commons «С указанием авторства — С сохранением условий» (CC BY-SA); в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Фонд Викимедиа (Wikimedia Foundation, Inc.)