

Технические особенности и новаторские решения процессоров AMD

Курсы лекций «ЭВМ и периферийные
устройства»

Факультатив

Лектор - доцент, к.т.н.

Кузьмин Александр Александрович

Часть 1. AMD K6: история успеха

Факты предыстории.

Длительное время компания Advanced Micro Devices (AMD), подобно фирме Cyrix, производила центральные процессоры 286, 386 и 486, которые были основаны на разработках Intel.

Попытки существенно потеснить Intel на рынке процессоров предпринимались. K5 был первым независимо созданным x86 процессором, на который AMD возлагала большие надежды, но они не оправдались.

Ситуация изменилась весной 1996 года, когда компания AMD приобрела калифорнийскую фирму-конкурента NextGen. Серия MMX-совместимых процессоров K6 (ранее - Nx686, переименована после приобретения NextGen) была запущена в середине 1997 года, за несколько недель до Cyrix 6x86MX, и сразу была одобрена пользователями.

Успех: технические особенности и новаторские (на момент создания) свойства

В середине 1998 года началось производство процессоров AMD K6-2 с 9,3 миллионами транзисторов на кристалле по 0,25-микронной технологии. Процессор имел инновационную эффективную микроархитектуру **RISC86**, большой (64 Кбайт) кэш первого уровня (двухпортовый кэш данных на 32 Кбайт, кэш команд на 32 Кбайт с дополнительной кэш-памятью на 20 Кбайт для предварительного дешифрирования), а также улучшенный модуль работы с плавающей запятой.

Самое новое и существенное

Производительность процессоров шестого поколения.

Высокая **эффективность при работе с операционными системами Windows**.

Шестикомандная суперскалярная **микроархитектура RISC86**, включающая в себя

- а) семь параллельных исполнительных блоков;
- б) множественные сложные дешифраторы команд x86/RISC86;
- в) двухуровневое предсказание ветвлений;
- г) «спекулятивное» (по предположению) исполнение команд;
- д) полное исполнение команд с изменением последовательности;
- е) переименование регистров при передаче данных.

Большая (64 Кбайт) L1 кэш-память на кристалле

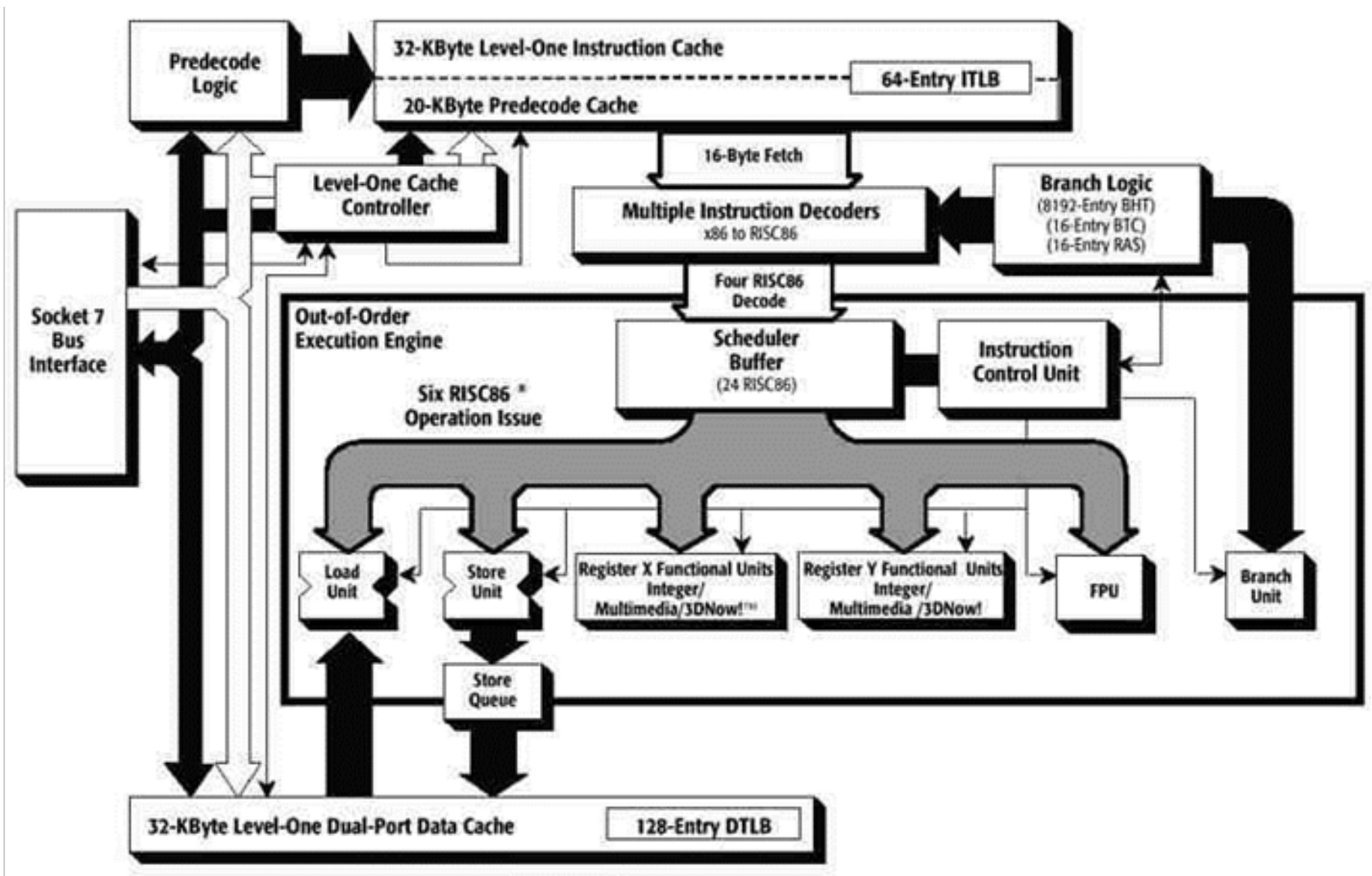
- а) 32 Кбайт кэш-память инструкций + кэш-память предварительного декодирования;
- б) 32 Кбайт двухпортовая кэш-память данных с обратной записью;
- в) протокол MESI для согласования блоков кэш-памяти.

Быстродействующее IEEE754-совместимое **устройство для вычислений с плавающей точкой (FPU)**.

Ставшие к моменту создания промышленным стандартом **мультимедийные расширения (MMX)**.

Полностью **Intel-совместимый режим управления** системой.

Микроархитектура процессора



Микроархитектура процессора (продолжение)

Отличительной особенностью микроархитектуры RISC86 процессора AMD K6-2 явилось внедрение суперскалярной конструкции с отдельным декодированием/исполнением, которая обеспечивает улучшенную производительность и полную совместимость с x86 бинарным программным обеспечением.

Применённая технология включает декодирование множественных x86 инструкций в одноктактовые RISC операции, их исполнение с изменением последовательности, передачу данных, «спекулятивное» (по предположению) исполнение и переименование рабочих регистров. Процессор содержит параллельные дешифраторы, централизованный планировщик RISC86-операций, а также семь исполнительных блоков, которые обеспечивают суперскалярное исполнение x86-инструкций.

Микроархитектура процессора (продолжение)

Очень важно!!

Новаторство заключено в том, что процессор выполняет внутреннее декодирование x86-инструкций в RISC операции, сохраняя при этом верность таким принципам RISC систем, как кодирование фиксированной длины, регулируемые поля инструкций и большой набор рабочих регистров.

Вместо того, чтобы напрямую выполнять сложные x86-инструкции, длина которых может составлять от 1 до 15 байт, процессор AMD K6-2 выполняет менее сложные RISC86 операционные коды фиксированной длины.

Логика предсказания ветвлений, применённая в AMD K6-2, включает таблицу истории ветвлений из 8192 статей, кэш адресатов ветвлений и обратный адресный стек. Это в совокупности обеспечивает вероятность правильного предсказания, превышающую 95%.

Микроархитектура процессора (продолжение)

Декодирование x86-инструкций начинается ещё до их внесения в размещённую на кристалле кэш-память инструкций. Логика предварительного декодирования определяет длину x86-инструкции на битовой основе. Информация предварительного декодирования хранится вместе с x86-инструкциями в кэш-памяти для последующего использования дешифраторами. Команды подразделяются на три категории согласно трём типам декодирования:

- «короткие» дешифраторы – декодируют наиболее часто используемые x86-инструкции;
- «длинные» дешифраторы – декодируют менее часто используемые команды;
- векторные дешифраторы – декодируют редко встречающиеся и сложные x86-инструкции.

Микроархитектура процессора (продолжение)

Централизованный планировщик (Scheduler) и его буфер управляются блоком управления командами (Instruction Control Unit – ICU). Блок ICU обслуживает выполнение до 24-х RISC86 операций. Размеры буфера, вмещающего 24 операции оптимально для эффективного использования шестиступенчатого RISC86 конвейера и семи параллельных исполнительных блоков процессора.

Блок ICU может одновременно выдавать на исполнительные блоки до шести RISC86 операций за один такт.

Управляя RISC86 операциями, планировщик использует 48 физических регистров, содержащихся в микроархитектуре RISC86. Эти регистры находятся в общем регистровом файле и объединены в группы: 24 общих регистра и 24 регистра переименования.

Микроархитектура процессора (продолжение)

Чтобы свести к минимуму задержки, связанные с распространёнными в программном обеспечении x86 инструкциями ветвления, процессор AMD K6-2 использует динамическую логику ветвлений.

Динамическая логика ветвления содержит таблицу истории предсказаний ветвлений, кэш адресатов ветвлений и обратный стек. Схема предсказания организована как двухуровневая. Поскольку таблица истории ветвлений не хранит предсказаний целевых адресов, специальные адресные АЛУ вычисляют целевые адреса в процессе декодирования инструкции. Кэш адресатов ветвлений обеспечивает более точное предсказание, помогая избежать дополнительного такта при выборе адресата. Этот специальный кэш поставляет на дешифраторы первые 16 байт инструкций адресатов при предсказании ветвлений.

Кэш-память и предварительное декодирование

L1-кэш состоит из двух независимых блоков: L1-кэша данных (Level-One Dual Port Data Cache) и L1-кэша инструкций (Level-One Instruction Cache) с кэшем предварительного декодирования (Predecode Cache).

L1-кэш данных предназначен только для хранения данных и имеет объем 32 Кбайт.

Сложнее организован L1-кэш инструкций: наряду с инструкциями, для хранения которых предназначены 32 Кбайт памяти, в нем хранятся так называемые "биты преддекодирования" (predecode bits) — для них отведено 20 Кбайт памяти. После загрузки инструкции в L1-кэш инструкций выполняется ее предварительное декодирование (predecoding) и **к каждому байту инструкции добавляется пять бит** (из этого и следует соотношение 32 Кбайт/20 Кбайт = 8/5), в которые записывается информация о количестве байт, оставшихся до начала следующей инструкции. Эта информация используется на этапе декодирования x86-инструкций в RISC86-инструкции.

Кэш-память и предварительное декодирование (продолжение)

Используемая в процессоре AMD K6-2 конструкция кэш-памяти имеет секторную организацию. Каждый сектор состоит из 64 байтов, сконфигурированных как две 32-байтовые строки, у которых общий тег, но отдельные пары битов протокола MESI. Суть в том, что каждая строка кэш-памяти, согласно протоколу MESI, может находиться в одном из четырех состояний:

- Modified – строка в кэш-памяти была изменена (отлична от основной памяти) и новое значение доступно только в данной кэш-памяти;
- Exclusive – значение строки совпадает со значением в основной памяти, но его (значения) нет ни в одной другой кэш-памяти;
- Shared – значение строки совпадает со значением в основной памяти и может присутствовать в другой кэш-памяти;
- Invalid – строка кэш-памяти содержит недостоверную информацию.

После того, как L1-кэш инструкций полностью заполнится данными, инструкции вместе с преддекодированными битами передаются в буфер инструкций (Instruction Buffer).

Модуль декодирования

Модуль (Multiple Instruction Decoders) извлекает x86-инструкции (до 16 байт данных с инструкциями за один такт) с битами преддекодирования из буфера инструкций (Instruction Buffer), определяет границы инструкций и преобразует их в RISC86-инструкции.

Непосредственно преобразованием занимаются четыре декодера: два для декодирования простых (Short Decoder #1, Short Decoder #2) и два для декодирования сложных x86-инструкций (Long Decoder, Vector Decoder). Одновременно могут работать либо два декодера Short Decoder #1 и Short Decoder #2, либо декодер Long Decoder, либо декодер Vector Decoder.

Модуль декодирования (продолжение)

Два декодера Short Decoder #1 и Short Decoder #2 работают параллельно и обрабатывают наиболее часто используемые x86-инструкции — move, shift, branch, операции ALU и FPU, а также инструкции из наборов команд MMX и 3DNow!.

Декодеры обрабатывают только часто используемые (most commonly-used) x86-инструкции длиной **не более семи байт**.

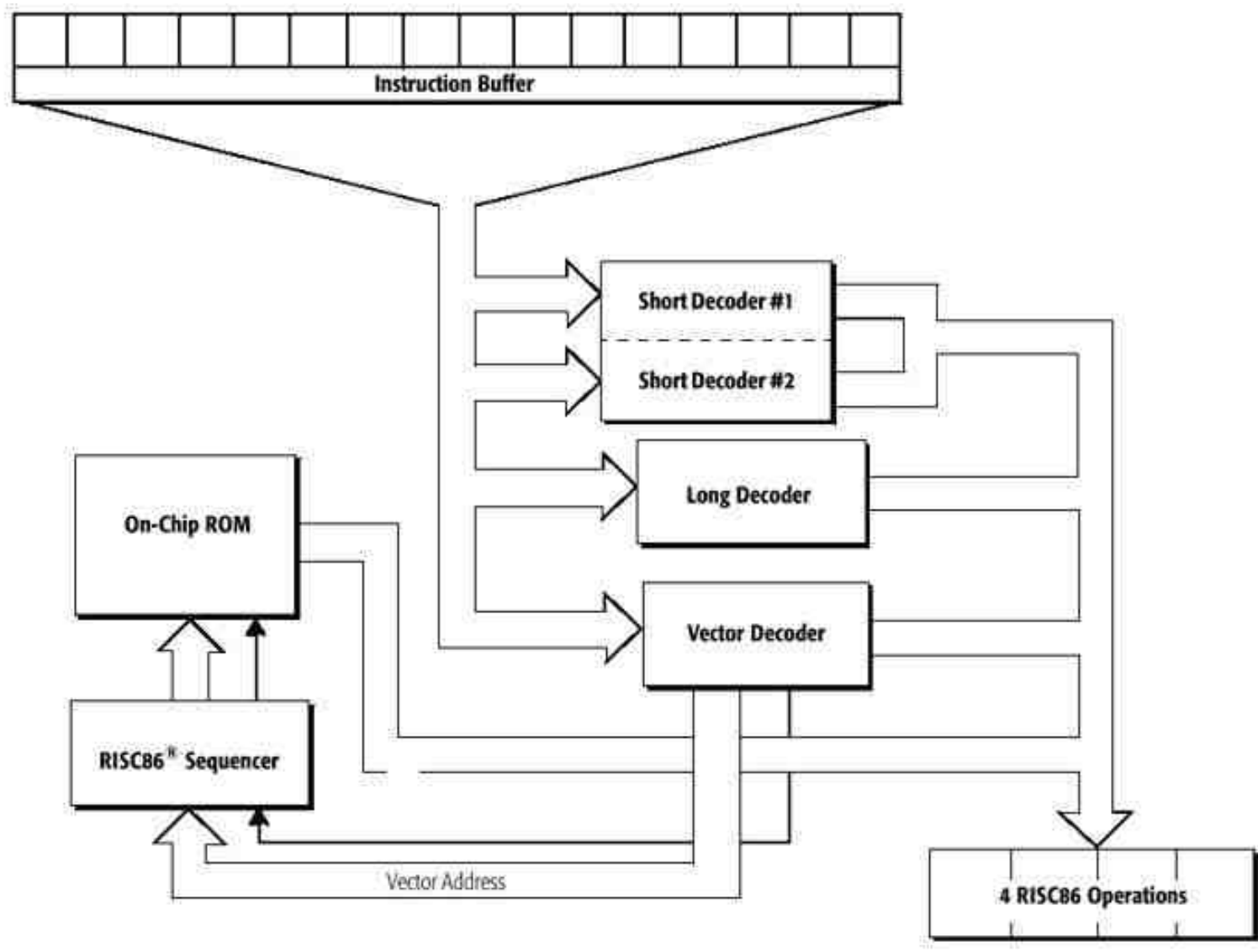
Каждый может преобразовать только одну такую x86-инструкцию и сгенерировать 0 (например, при обработке x86-инструкции NOP), одну или две RISC86-инструкции за такт. Таким образом, за один такт оба декодера могут сгенерировать до 4-х RISC86-инструкций.

Модуль декодирования (продолжение)

Редко используемые инструкции (semi-commonly-used) длиной до семи байт и обычные инструкции (commonly-used) с длиной большей семи байт, но **меньшей или равной 11 байтам** обрабатываются декодером Long Decoder, который может декодировать только одну такую x86-инструкцию и сгенерировать до 4-х RISC86-инструкций за такт.

Все остальные преобразования (более сложные инструкции, прерывания и т.д.) выполняются декодером Vector Decoder. В этом случае Vector Decoder генерирует набор первых RISC86-инструкций и адрес заранее predetermined набора последующих инструкций, который хранится в ROM-памяти (On-Chip ROM) и извлекается блоком RISC86 Sequencer.

Структурная схема модуля декодирования процессора



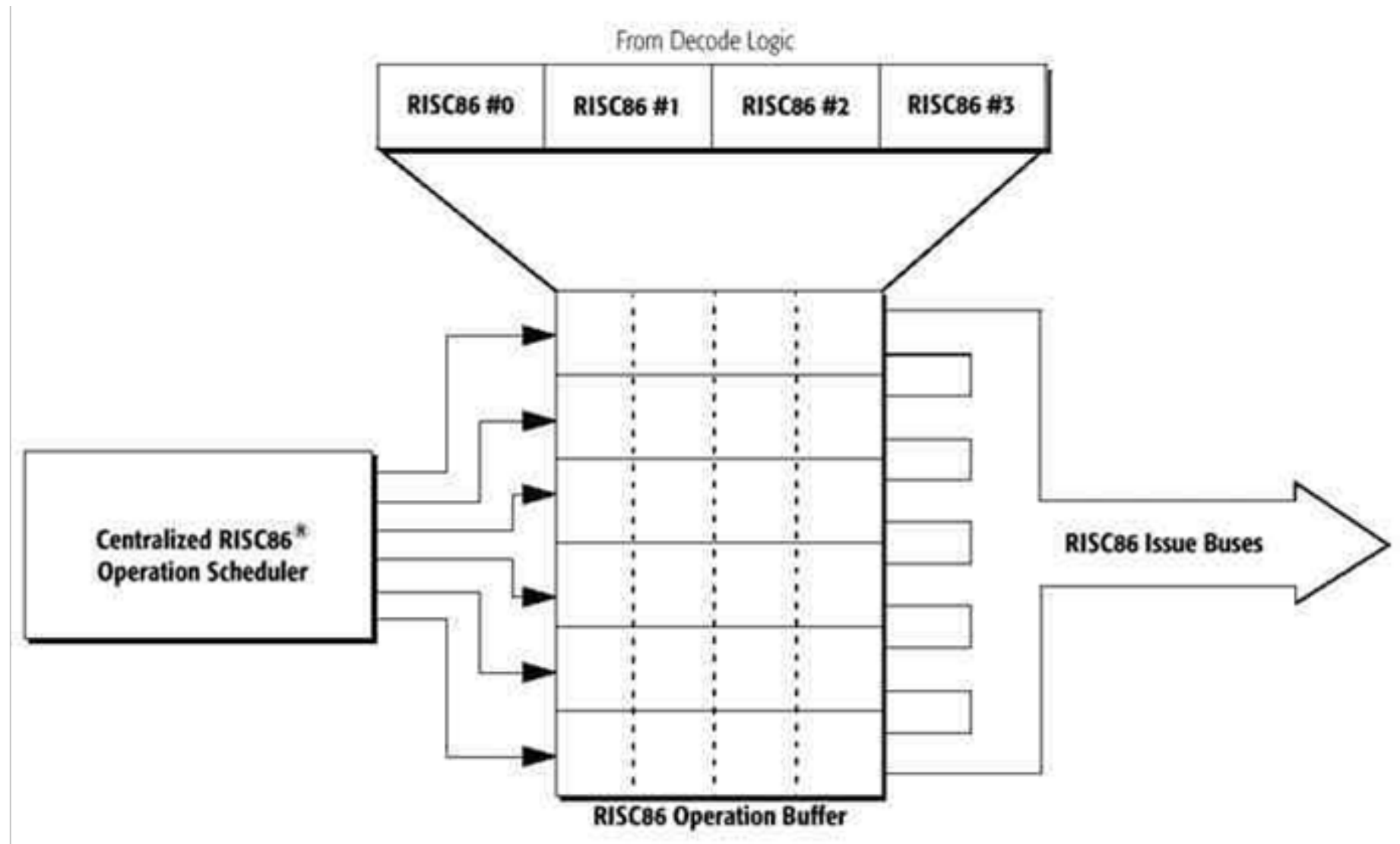
Все наборы RISC86-операций, генерируемые декодерами и извлекаемые из On-Chip ROM всегда (!) состоят из групп, содержащих по четыре RISC86-операции. В том случае, если их получилось меньше, недостающее количество заполняется пустыми RISC86-инструкциями NOP. Например, если Long Decoder преобразовал x86-инструкцию в три RISC86-инструкции, то к ней добавляется одна RISC86-инструкция NOP. Получившийся поток из таких групп поступает в буфер планировщика (Scheduler Buffer) — за один такт всегда передается группа из четырех RISC86-операций.

Центральный планировщик

Планировщик (Centralized RISC86 Operation Scheduler)— это «сердце» процессора AMD-K6-2. Он следит за процессом исполнения RISC86-инструкций, приведением результата их исполнения к x86-архитектуре, а также возвращением результатов «спекулятивного» выполнения x86-инструкций в соответствии с их порядком поступления на вход процессора.

В буфере планировщика может одновременно содержаться до 24-х RISC86-инструкций. Любая из них может быть в любой момент передана на исполнение соответствующему вычислительному блоку (store, load, branch, register X integer/multimedia, register Y integer/multimedia, floating-point), если, конечно, блок свободен. Таким образом, реализуется исполнение инструкций в порядке, отличном от порядка их поступления в буфер (out-of-order execution). В общей сложности планировщик может передать на выполнение шесть и завершить (retire) также шесть RISC86-инструкций за такт.

Структурная схема центрального планировщика



Вычислительные блоки

Процессор AMD-K6-2 содержит 10 параллельных вычислительных блоков (Execution Units) — Store Unit, Load Unit, Integer X ALU, Integer Y ALU, MMX ALU (X), MMX ALU (Y), MMX/3DNow! Multiplier, 3DNow! ALU, FPU и Branch Unit. Каждый блок работает независимо от остальных, так что несколько блоков могут обрабатывать переданные им на исполнение RISC86-инструкции одновременно.

Integer, MMX - и 3DNow!-инструкции передаются по двум независимым шинам — Register X Issue Bus и Register Y Issue Bus. При этом блоки Integer X ALU и MMX ALU (X) подключены только к шине Register X Issue Bus, а Integer Y ALU и MMX ALU (Y) — только к шине Register Y Issue Bus. А вот блоки MMX/3DNow! Multiplier и 3DNow! ALU подключены сразу к обеим шинам, как и блок MMX Shifter, функция которого заключается в том, чтобы переключать блоки MMX/3DNow! Multiplier и 3DNow! ALU между шинами.

На схеме на следующем слайде показаны вычислительные модули групп Register X и Register Y процессора AMD -K6-2

