

7 билет

Процессорные архитектуры ч.1: Наборы инструкций x86, x86-64.

Под архитектурой процессора обычно понимают две совершенно разные сущности.

С программной точки зрения архитектура процессора — это совместимость с определённым набором команд (Intel x86), их структуры (система адресации, набор регистров) и способа исполнения (счётчик команд).

С аппаратной точки зрения архитектура процессора — это некий набор свойств и качеств, присущий целому семейству процессоров (Skylake – процессоры Intel Core 5 и 6 поколений).

Архитектура x86

Данная архитектура является представителем CISC. Под x86 имеют в виду как архитектуру процессора, так и одноименный набор команд.

Впервые реализована компанией Intel. Название образовано от двух цифр, которыми заканчивались названия процессоров Intel ранних моделей. Помимо Intel была реализована и другими производителями (например, AMD входит в их число). В каталогах с файлами для архитектур она зачастую называется "i386".

Изначально архитектура разрабатывалась как 16-разрядная, но очень скоро был произведен переход на 32-разрядные регистры и шины. Изначально реализована только сегментная организация памяти, позднее появилась поддержка страничной организации.

x86 поддерживает весьма большое количество наборов инструкций, которые можно разделить на:

1. Команды общего назначения;

Различные команды передачи данных (сравнения, преобразования и т.д.), арифметические команды для целых чисел и с плавающей точкой, логические команды, побитовые сдвиги и вращения, локальная работа с битами (установка, сброс и пр.), команды передачи управления (для различных обработок прерываний и т.д.), работа со строками (последовательностями), работа с сегментными регистрами.

2. Системные команды;

Команды для работы с регистрами, остановка процессора и подобное.

3. Команды управления состоянием сопроцессора и SIMD;

сохранение и восстановление состояния сопроцессора и SIMD.

4. Команды технологии MMX (Multimedia Extensions);

Изначально предназначен для ускорения процессов кодирования-декодирования потоковых аудио и видеоданных. Основан на выполнении действий сразу над несколькими наборами характерных данных (SIMD-расширение).

5. Команды расширения SSE (2,3..);

Набор инструкций для реализации парадигмы SIMD.

6. Команды расширения 3DNow!;

Набор инструкций для потоковой обработки вещественных чисел одинарной точности. С некоторого момента расширение не поддерживается.

7. Команды сопроцессора (x87);

специальный набор инструкций для работы с математическими вычислениями. Название образовалось из-за того что изначально выпускались отдельные математические сопроцессоры и их названия заканчивались цифрами 87. Отличаются от стандартного математического набора команд использованием более широких регистров.

8. Далеко не сразу появился **набор команд AVX (2)**; Набор команд в основном представляющий из себя улучшение и переосмысление наборов команд SSE.

Полный список команд (почти полный, отсутствуют SSE 4,5 и AVX 1, 2):

<http://www.ccf.it.nyu.edu/~kireev/lab2/lab2com.htm>

Архитектура x86-64

Данная архитектура ознаменовала переход процессоров с набором команд x86 к 64-разрядным архитектурам. Данная архитектура является расширением архитектуры x86 с почти полной обратной совместимостью. В каталогах с файлами для архитектур она зачастую называется "amd64".

Рассматривая количество поддерживаемых команд можно наглядно понять в чем суть CISC-архитектур.

Вообще говоря рассмотренные архитектуры являются почти свободно поставляемыми, однако для доступа к некоторым возможностям требуется приобретать лицензию у Intel.

8 билет

Процессорные архитектуры ч.2: Наборы инструкций SPARC, Power.

Архитектура SPARC (Scalable Processor ARChitecture)

Это RISC-архитектура, первоначально разработанная компанией Sun Microsystems. Данная архитектура распространяется не свободно, лицензия приобретается у компании SPARC International. Изначально разрабатывались 32-разрядные процессоры, однако в последующем был произведен переход на 64-разрядную архитектуру.

Наборы инструкций можно разделить на 6 основных категорий:

1. **Загрузка / сохранение данных;**

Этот набор инструкций является единственным, который имеет доступ к адресному пространству. Способен оперировать такими типами как: байт (8 бит), пол слова(16 бит), слово (32 бит) и двойное слово (64 бит).

2. **Арифметические и логические операции, операции побитного сдвига;**

Данные операции делают требуемые вычисления и записывают результат в регистр.

3. **Передача управления;**

Данные инструкции обеспечивают контроль событий таких как прерывания и переходы (jump).

4. **Чтение / запись контрольного регистра;**

Позволяют пользовательским процессам редактировать видимые им регистры.

5. **Работа с плавающей точкой;**

Эти инструкции представляют из себя инструкции напоминающие описанные в пункте 2, однако работают с числами с плавающей точкой.

6. **Работа сопроцессора;**

Данные инструкции определяются сопроцессором, если такой имеется. Стандарты инструкций сопроцессора, которые поддерживаются: это CPop1 и CPop2.

Полное описание инструкций в мануале: <https://www.gaisler.com/doc/sparcv8.pdf>

Архитектура Power.

Это RISC-архитектура разработанная и развиваемая компанией IBM. Название позже было расшифровано как Performance Optimization With Enhanced RISC (оптимизация производительности на базе расширенной архитектуры RISC). Начиная с POWER3 микропроцессоры поддерживают полный набор команд 64-битной архитектуры PowerPC и не поддерживают старые команды, которые были удалены из архитектуры одновременно с появлением стандарта PowerPC. Также прекращена поддержка расширенных инструкций POWER2.

Набор команд был основан на идее суперскалярной обработки. В базовой архитектуре команды распределяются по трем независимым исполнительным устройствам: устройству переходов, устройству с фиксированной точкой и устройству с плавающей точкой. Команды могут направляться в каждое из этих устройств одновременно, где они могут выполняться одновременно и заканчиваться не в порядке поступления. Для увеличения уровня параллелизма, который может быть достигнут на практике, архитектура набора команд определяет для каждого из устройств независимый набор регистров.

Архитектура POWER расширена несколькими "смешанными" командами для сокращения времен выполнения. Возможно единственным недостатком технологии RISC по сравнению с CISC, является то, что иногда она использует большее количество команд для выполнения одного и того же задания. Было обнаружено, что во многих случаях увеличения размера кода можно избежать путем небольшого расширения набора команд, которое вовсе не означает возврат к сложным командам, подобным командам CISC. Например, значительная часть увеличения программного кода была обнаружена в кодах пролога и эпилога, связанных с сохранением и восстановлением регистров во время вызова процедуры. Чтобы устранить этот фактор IBM ввела команды "групповой загрузки и записи", которые обеспечивают пересылку нескольких регистров в/из памяти с помощью единственной команды. Соглашения о связях, используемые компиляторами POWER, рассматривают задачи планирования, разделяемые библиотеки и динамическое связывание как простой, единый механизм. Это было сделано с помощью косвенной

адресации посредством таблицы содержания (TOC - Table Of Contents), которая модифицируется во время загрузки. Команды групповой загрузки и записи были важным элементом этих соглашений о связях.

Архитектура POWER обеспечивает также несколько других способов сокращения времени выполнения команд такие как: обширный набор команд для манипуляции битовыми полями, смешанные команды умножения-сложения с плавающей точкой, установку регистра условий в качестве побочного эффекта нормального выполнения команды и команды загрузки и записи строк (которые работают с произвольно выровненными строками байтов).

Мануал: http://math-atlas.sourceforge.net/devel/assembly/ppc_isa.pdf

И что-то еще, вроде бы по теме:

http://citforum.ru/database/skbd/glava_23.shtml