[Последнее добавление]
[Оффлайн копия сайта]

18 Мая 2006. Открытие сайта.
Привет. На этом сайте размещается информация о Nintendo Gamecube (далее GC), архитекутра которого основана на процессоре PowerPC. Сайт предполагает знание читателем компьютерной техники (железа) и языков программирования (особенно Си).

Цели, которые ставятся перед данным ресурсом:
- Полное рассмотрение архитектуры GC и его центрального процессора;
- Реверс-инжиниринг программного обеспечения для GC (SDK, устройство игр, итд);
- Написание различных программ для куба;
- Создание программного эмулятора GC;

Вся представленная информация законна. Данный сайт НЕ распространяет пиратскую продукцию или нелицензионные копии программ и их исходных кодов.

В первую очередь я хочу познакомить читателя с Gamecube:

GC (кодовое имя Dolphin) был разработан корпорацией Nintendo. При создании консоли Nintendo привлекла много компаний, которые разработали для нее различные аппаратные узлы:
Macronix. Эта малоизвестная компания разработала для куба периферийную шину, так называемый EXI - Expansion Interface ('Интерфейс расширений'). Если сравнивать с компьютером, то эта шина является аналогом USB. Устройства которые подключены к шине EXI - тоже разработка Macronix. Этими устройствами являются карты памяти, сетевой адаптер (Broad-Band Adapter, BBA), а также чип, содержащий загрузочный ROM и часы реального времени на батарейке. Ещё Macronix разработала для N сигнальный процессор для обработки звука (DSP).
MoSys. Разработала для куба чип оперативной памяти 1T-SRAM.
ArtX. Компания ArtX разработала для N графический процессор (кодовое имя Flipper). Впоследствии эта компания была куплена ATI.
S3. В графическом процессоре куба используется специальный формат сжатия текстур - S3 Texture Compression (S3TC).
Matsushita (Panasonic). Эта компания разработала привод для чтения дисков формата mini-DVD. Уникальной особенностью привода является многоуровневая система защиты дисков от копирования, которая тем не менее была успешно сломана (на что потребовалось 4 года). Некоторое время Panasonic производил модифицированную версию GC под названием "Panasonic Q". Различия заключались только в DVD-приводе и возможно прошивке микроконтроллера (для поддержки проигрывания обычных DVD). Аппаратно это был GC NTSC-версии + DVD-проигрыватель, который воспринимал либо только кубовские защищенные мини-DVD, либо обычные DVD.
IBM. Разработала центральный процессор (кодовое имя Gekko), основанный на архитектуре PowerPC. Gekko полностью совместим с 32-разрядной версией архитектуры PowerPC и в добавок содержит несколько дополнительных наворотов, для улучшения производительности. Прототипом для Gekko послужила 750-я модель (G3).

Таким образом Nintendo фактически не создала для куба ни одного "умного" чипа. Инженерам осталось только разместить все компоненты на материнской плате, и разработать дизайн корпуса. Однако это нисколько не умаляет Nintendo, так как при создании куба она выбрала очень удачное архитектурное решение, при этом используя минимум средств для получения максимальной производительности. Оригинальными разработками N являются:
Контроллер - эволюционное развитие контроллера Nintendo 64 (который похоже совместим с кубом на аппаратном уровне (?)).
Видео и аудио DAC - для вывода видеосигнала в форматах PAL/NTSC/SCART.
Также N провела большую работу при создании программного обеспечения для разработки игр (SDK). Программировать для куба действительно просто и удобно, хотя не так легко стать официальным разработчиком и получить SDK. Возможно это и повлияло на то, что куб так не популярен у разработчиков, и соответсвенно обычных игроков, которые смотрят в сторону более доступной PS2.

Вот спецификации куба (не те, которые приводит N публике, а более точные, проверенные исследованием SDK):

Архитектура. На материнской плате расположены: CPU Gekko, ASIC Flipper, два чипа оперативки (RAM) по 12 мб каждый, а также дополнительная память (ARAM) размером 16 мб (всего системной памяти - 40 мб). Внутри Flipper'a находится всё железо, кроме tv-out видеосистемы, которая расположена на материнской плате. Характеристики шины: 32 разряда на адресную, 64 разряда на шину данных. Частота шины: 162 мгц (1/3 от частоты CPU). Пропускная способность 1.3 гига в секунду. Дополнительная память (ARAM) имеет 8-разрядную шину, работает на 81 мгц (1/2 от системной шины).

Пропускная способность ARAM - 81 мегабайт в секунду. Доступ к ARAM осуществляется через DMA-канал. ARAM используется как временное хранилище для аудио и графических данных.

Микропроцессор (CPU). Модифицированный IBM PowerPC 750. Кодовое имя "Gekko". Характеристики: 0.18 медь (рассеиваемая мощность всего 5 вт), размер кристалла ~40 мм2, 21 млн. транзисторов. Тактовая частота 486 мгц. Указанная производительность 1125 mips/10.5 gflops, на самом деле сравним с 900-1000 мгц Pentium. Кэш L1: 32 KB инструкции, 32 KB данные, L2: 256 KB данные. Отличительные особенности:
- Расширенный набор инструкций Paired Single - аналог SIMD MMX/SSE.
- L1 кэш данных может работать в режиме блокировки.
- Буфер подкачки (write gather buffer).
- Большой L2 кэш данных.

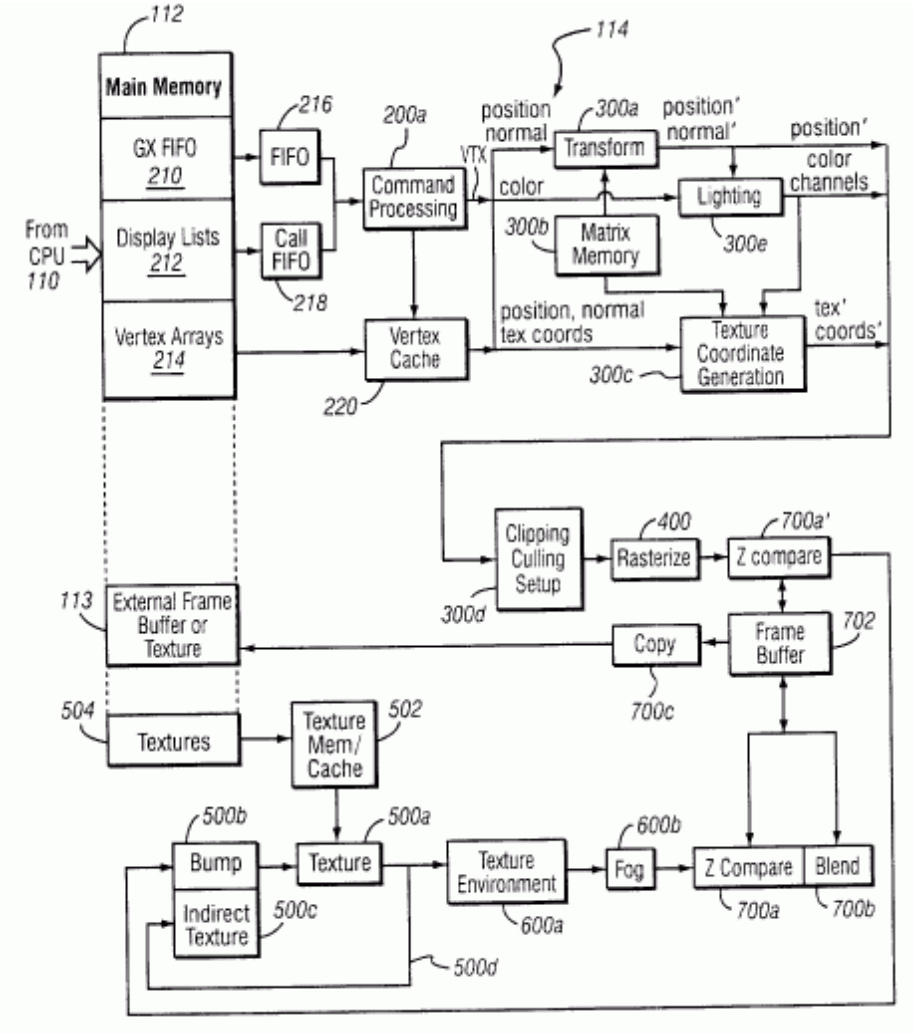
Видеосистема (TV-Out). Поддержка видеорежимов PAL, NTSC, M-PAL (Бразилия), PAL60 (европейский Scart). Развертка: черезстрочная (interlaced), построчная (progressive). Поддержка DTV (?), стерео-разделения кадра (для "3D"-очков), светового пистолета. Полноэкранный антиалиасинг конфигурируемый tap-фильтрами. Формат видеобуфера Y1UY2V (две RGB-точки (6 байт) пакуются в одну "спаренную" YUV (4 байт), за счёт этого сохраняется полоса пропускания). Видеобуфер (для tv-out) располагается в оперативной памяти (RAM).

Носитель данных. 3" DVD, разработанный по технологии Matsushita. Защищён от копирования нестандартными bar-кодами и кроме того данные секторов зашифрованы. Емкость диска 1,459,978,240 байт. Диск вращается с постоянной угловой скоростью (CAV). Скорость чтения 2-3 мб в секунду (сравним с 13X CDROM). Есть поддержка потокового DVD-Audio в формате ADPCM, 48 кгц. Привод является интеллектуальным устройством (содержит микроконтроллер MN 102000).

Далее характеристики железа, которые входят в состав ASIC Flipper'a:

Звуковая система. Один DMA канал, переключаемая частота звука - 32/48 кгц. Максимальный размер аудио-буфера - 1 MB. Аудио-буфер располагается в оперативной памяти (RAM). Также в состав Flipper входит сигнальный процессор (DSP), для обработки звука. Этот процессор позволяет смешивать до 64 (и более) звуковых каналов (причем в 3D) с эффектами эхо, реверберации, ADSR, для каждого канала. С помощью аудио-библиотек можно использовать любые мыслимые аудио-эффекты. При этом DSP работает параллельно графическому и центральному процессору, за счет чего не происходит падения в производительности. Архитектура DSP неизвестна (в последнее время проводятся попытки узнать о ней больше). Синхронизация между CPU и DSP, а также передача команд осуществляется путем посылки "сообщений". Для временного хранения аудио-данных используется дополнительная память ARAM.

Графический процессор. Разработка компании ArtX. Позднее ArtX была куплена ATI. Вклад ATI - красный стикер на приставке. Графический процессор можно условно разделить на 3 части: командный процессор, геометрический процессор и растеризатор.
Командный процессор: аппаратный FIFO-буфер для параллельного выполнения графических команд (пока CPU передает команды, графический процессор рисует сцену). Кэш для вершин. Дисплейные списки (вложенность не допускается). Массивы вершин (координаты вершин не передаются через FIFO, а находятся в оперативке), причем формат вершин может быть разным - не только вещественные числа, но и числа с фиксированной точкой.
Геометрия: геометрические вычисления производятся числами с фиксированной точкой. 8 аппаратных RGBA-источников света с диффузной и бликовой светимостью. (diffuse/specular lights). Изменение яркости по углу и расстоянию (angle/distance attenuation). Toon shading, bump mapping.
Растеризатор: 24 разрядный цвет (RGB), глубина Z-буфера 24-бит. Встроенный графический видеобуфер - 2 MB, отсюда следует разрешение встроенного буфера - 640x528 пикселей. Вся отрисовка ведется во встроенный видеобуфер, затем копируется во внешний видеобуфер, для TV-Out, который располагается в оперативке. Память для текстур - 2 MB. Кэш для текстур. Форматы текстур: индексные TLUT, RGB565, RGB5A3, RGBA, IA4/8, S3TC (сжатие). Альфа-канал. Трёхмерные текстуры. Мультитекстурирование (до 16 текстур, 8 оригинальных). Текстурирование по текстуре (indirect texturing). MIP-mapping, билинейная/трилинейная фильтрация, анизотропная фильтрация (max=4). Максимальный размер текстуры 1024x1024. Несколько типов тумана, blending. Gamma-коррекция.
Короче говоря в графическом процессоре куба есть всё, кроме шейдеров. Вообще говоря в графическом чипе нет вычислительного устройства (т.е. какого-либо процессора), он является тем, что называется английским термином "state-machine", а по нашему - вычислитель с жесткой логикой. Естественно это не позволило ArtX реализовать технологию шейдеров. Указанная производительность - 6-12 млн полигонов в секунду (100-200 тыс. треугольников в кадре), учитывая все игровые условия (освещение, текстуры, звук и пр.).



Схематическое изображение графического процессора GC.

Прочее:

Устройства ввода. 4 аналоговых контроллера с авто-калибровкой и моторчиками. Две аналоговые ручки, два аналоговых триггера, 6 кнопок, крестовина. Существуют множество вариации стандартного контроллера, в том числе и беспроводные.
Периферия. Сетевой адаптер (BBA), модем, Gameboy Advance, который подключается вместо контроллеров, и используется для включения дополнительных фишек в играх. Два слота для карт памяти.

Питание. AC Adapter DC12V x 3.5A
Размеры. 4.3" (в) x 5.9" (ш) x 6.3" (д)

Ресурсы по статье.
- Исследование разъема контроллера GC
- Исследование разъема контроллера N64
- Разработка для GC (основной зарубежный сайт по GCDEV)
- Dextrose (Декстроза)
- Список ссылок на основные зарубежные ресурсы о GCDEV.
- Wikipedia о GameCube (немного неточные спецификации)
- Патент США 6,609,977 - "External interfaces for a 3D graphics system".

[Обсудить в гостевой]

---

29 Мая 2006. Операционная система GC (Dolphin OS).
Dolphin OS является разработкой Nintendo. Операционная система создавалась с 1998 по 2001, после чего первая стабильная версия была выпущена вместе с SDK.

Dolphin OS основана на операционной системе SGI, которая была использована в Nintendo 64 (известна также, как "libultra"). Разработчики Dolphin OS просто переделали её под PowerPC (Gekko).

Также, как и свой предшественник, Dolphin OS является статически линкуемой библиотекой. Определить эту ОС можно так: однопользовательская, в памяти находится только один процесс (process), но поддерживается многозадачность (threads). В системе нет ядра (kernel) и все приложения запускаются с повышенными привилегиями OEA, т.е. программам доступна вся память и системные регистры. Такое решение является оправданным, потому что на GC выполняется только одна программа - игра.

В состав ОС входит не только непосредственно операцинная система, но и набор библиотек для создания полноценных программ для GC: аудио-библиотека, графическая библиотека, а также библиотеки для программирования железа GC и управления центральным процессором. Можно сказать, что ОС предоставляет полноценный контроль за железом куба, завернутый в красивую упаковку библиотек. Поэтому разработчику не приходится иметь дело с регистрами аппаратуры, но в то же время взаимодействие программ с железом происходит очень быстро, благодаря тому, что это встроено в ОС. Все вместе это дополняется различными утилитами, что и составляет GC SDK.

Если вас интересует, каким образом это сделано, то вот краткое описание:
- Программа для GC написана на Си.
- Набор библиотек Dolphin OS (os.a, ax.a, gx.a, ...) из SDK
- Всё вместе это компилируется в исполняемый ELF файл
- Затем конвертируется в исполняемый файл формата DOL
- Исполняемый файл DOL помещается на DVD вместе игровыми файлами (текстуры, модели, звуки и пр.)
- Образ DVD штампуется на заводах Nintendo, и продается потребителю.

Заключение: Dolphin OS нельзя назвать полноценой операционной системой в том понимании, к которому мы привыкли. Скорее всего, это удобный инструмент для создания игр.

[Обсудить в гостевой]

---

22 Августа 2006. Первые тесты.
Наконец-то я купил и скачал всё что нужно для полноценной разработки программ для GC. Схема выглядит следующим образом:

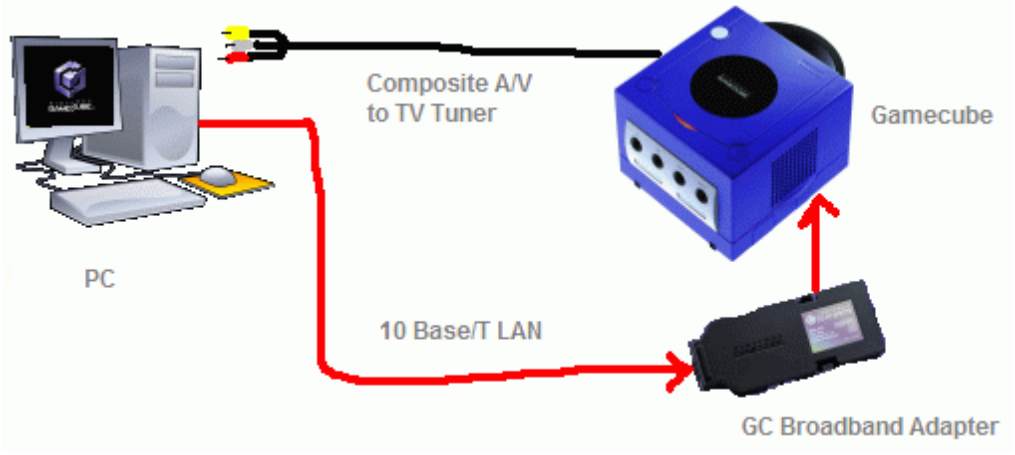
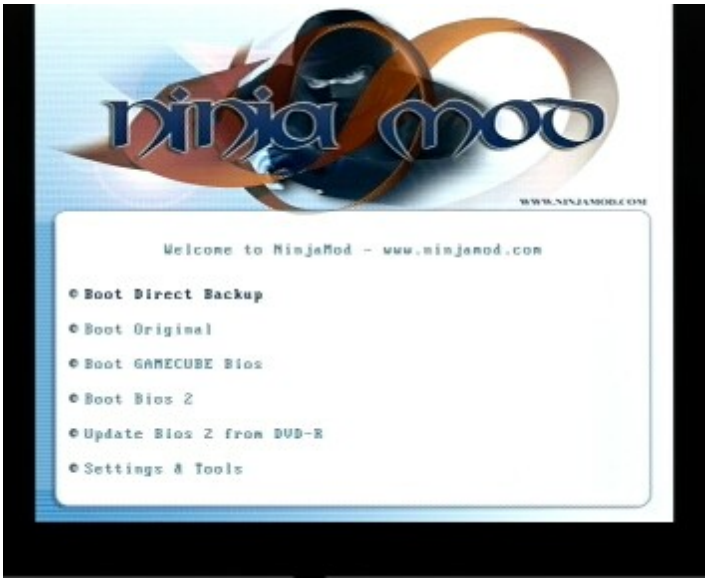


Схема подключения куба к компьютеру.

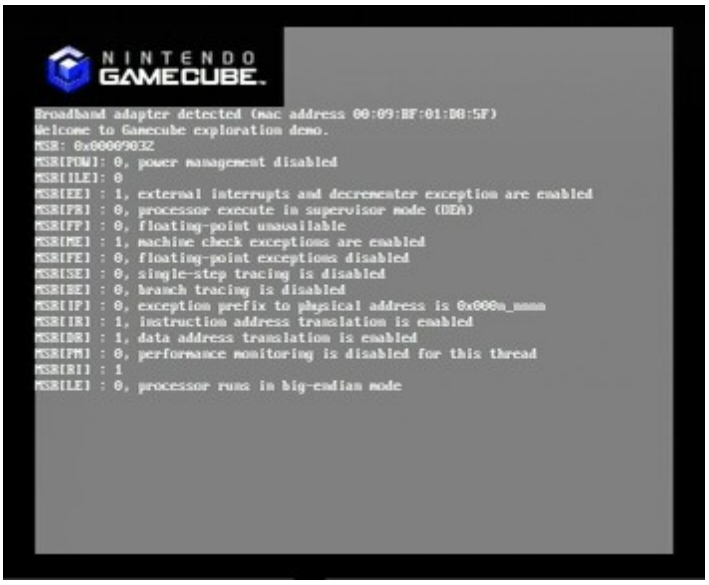
Аппаратная часть: чипованный куб DOL-001 PAL (есть возможность перепрошивки Boot ROM через USB), Broadband Adapter и соединение по локальной сети для загрузки программ. В компьютер установлен ТВ-тюнер для получения видеоизображения с куба прямо на мониторе.
Программная часть: NinjaMod, прошитый вместо оригинального IPL, который позволяет загружать исполняемые файлы (DOL, ELF) по сети, а также запускать DVD-R болванки. Для компиляции программ я использую Metrowerks CodeWarrior for GAMECUBE, а также Dolphin SDK, скачанные из пиринговых сетей. Для загрузки программ с компьютера я использую модифицированную программку NetCat. Для ТВ-тюнера используется программа FLY 2000 TV.
Хочу сделать замечание, что хотя кое-что из перчисленного нелегально, я не вижу своей вины, в том что я использую это в образовательных целях. Это не моя проблема, что Dolphin SDK и CodeWarrior утекли в сеть.

Процесс подготовки программы выглядит следующим образом: текст набирается и компилируется в среде CodeWarrior. Затем включается GC, и NinjaMod начинает "слушать" порт 4000.



Загрузка программы на выполнение производится командой NetCat: **nc.exe 192.168.1.32 4000 < "GCNDefault.dol"**

Все тесты я решил разделить на две части - тесты CPU Gekko и тесты Flipper-a. Первая программа, довольно простая. Я поставил цель организовать простейший вывод текста на экран для чтения диагностических сообщений. На этом скриншоте показано содержимое регистра Gekko MSR, после инициализации Dolphin OS:



По содержимому MSR можно выяснить в каком режиме работает Gekko при выполнении обычных программ для GC:
- Управление питанием выключено. Dolphin OS не использует режимы пониженного потребления мощности Gekko ("sleep", "doze").
- Процессор всегда работает в режиме супервизора. Т.е. обычные программы обладают правами ОС (особенность Dolphin OS).
- Порядок байт с которым работает Gekko - прямой (big-endian), в то время как у семейства X86 используется обратный порядок байт (little-endian). Это нужно учитывать при работе с файлами.
- При переключении контекста FPU используется хитрый трюк - "FPU N/A": регистры FPU сохраняются в контексте при переключении процесса не каждый раз, а только тогда, когда они изменяются. Это возможно благодаря особому исключению, которое возникает при попытке исполнения FPU-инструкции, если "FPU N/A" включен. Это позволяет сократить время на переключение (перегрузку) контекста.
Как видно даже такая нехитрая программка дает немало информации.

Ресурсы по статье.
- Первая версия программы для тестирования GC (исходный код).
- NinjaMod: бесплатная замена фирменного GC IPL.
- Lik-Sang: онлайн магазин которому можно доверять. Тут можно заказать GC и BBA.
[Печальная новость: Sony засудила Lik-Sang. Подробности на сайте. Где теперь купить GC и BBA?]
- MAXCONSOLE: здесь можно заказать ModChip для GC.
- FLY 2000 TV, единственная программа которая подошла к моему ТВ-тюнеру.
- NetCat

[Обсудить в гостевой]

---

11 Сентября 2006. Формат DOL.
Файлы с расширением **.dol** это исполняемые файлы, используемые операционной системой GC. Хотя все программы для GC компилируются в формат ELF, в состав SDK входит специальная утилита "makedol" для конвертирования ELF-файлов в DOL. Структура DOL-файла очень простая:

```
 --------------------
|    Загловок DOL    |
|      256 байт      |
|--------------------|
|Секции .text и .data|
|          .         |
|          .         |
|          .         |
 --------------------
```

В заголовке указывается информация о расположение (смещении) секций данных и кода, виртуальные адреса их загрузки в память, расположение секции BSS (статические данные), а также адрес точки входа в программу. Существует несколько ограничений:
- Секции должны быть выровнены на 32 байта (минимальный размер DMA пересылки с DVD)
- Рекомендуемый размер DOL-файла не более 4 MB.
- Количество секций кода до 7, секций данных до 11.
- Не допускается загружать что-либо выше адресов 0x81200000 (там находится программа начальной загрузки; DOL просто затрет загрузчик и система скорее всего зависнет).

Ресурсы по статье.
- Утилита DOLSplit 1.0, для работы с DOL-файлами (исходный код).
- Спецификации формата DOL (структура заголовка).
- Описание формата ELF.

[Обсудить в гостевой]

---

13 Октября 2006. Исследование регистров SPR.
Самый первый тест - узнать версию CPU, прочитав соотв. специальный (SPR) регистр - PVR. Значение PVR у Gekko **0x00083214**.

Далее я решил проверить все SPR регистры на их режимы (User/Supervisor, Read/Write). У PowerPC не все специальные регистры доступны в режиме пользователя, или например в режиме пользователя их можно только читать. Следующая таблица взята из официальной документации и содержит список всех SPR, их режимы и краткое описание.

Эти регистры определены Generic архитектурой PowerPC:

SPR	N	Доступ	Описание
XER	1	User: R/W Supervisor: R/W	Содержит флаги переполнения и переноса для арифметических операций, а также счетчик байт для инструкций lswx и stswx.
LR	8	User: R/W Supervisor: R/W	Link Register. Содержит адрес возврата в инструкциях перехода (branch).
CTR	9	User: R/W Supervisor: R/W	Count register. Используется в инструкциях перехода как счетчик циклов.
DSISR	18	User: -/- Supervisor: R/W	Используется в механизме исключений (содержит дополнительную информацию)
SRR0	26	User: -/- Supervisor: R/W	Содержит адрес возврата при исключении.
SRR1	27	User: -/- Supervisor: R/W	Содержит копию регистра MSR при исключении.
SPRG0-SPRG3	272-275	User: -/- Supervisor: R/W	Для внутреннего использования операционной системой.

BATs	528-543	User: -/- Supervisor: R/W	Используются в механизме блочной трансляции адреса (16 регистров: 8 DBATs, 8 IBATs)
SDR1	25	User: -/- Supervisor: R/W	Задает формат страниц в механизме виртуальной трансляции адреса.
EAR	282	User: -/- Supervisor: R/W	External access register. Используется в устаревшем способе доступа к аппаратным регистрам. В Gekko не используется, и все регистры отображаются на адреса физической памяти (hardware regs memory mapping)
DABR	1013	User: -/- Supervisor: R/W	Data address breakpoint. Точка останова на доступ к данным.
PVR	287	User: -/- Supervisor: R/-	Версия процессора.
DEC	22	User: -/- Supervisor: R/W	Decrementer. 32-разрядный счетчик. Считает вниз до 0, затем вырабатывается исключение. Используется для генерирования периодичных событий (например, объект OSAlarm в Dolphin OS)
TBL,TBU	284,285	User: -/- Supervisor: -/W	64-разрядный счетчик времени CPU. Состоит из двух 32-разрядных половинок (только для записи).

Следующие специальные регистры есть только у Gekko:

SPR	N	Доступ	Описание
DMAU,DMAL	922,923	User: -/- Supervisor: R/W	Используется для DMA-пересылки между памятью и locked L1 кэшем данных.
GQR0-GQR7	912-919	User: -/- Supervisor: R/W	Используются при преобразовании типов данных - в т.н. Quantization Load/Store инструкциях.
HID0	1008	User: -/- Supervisor: R/W	Для управления состоянием процессора и кэшей.
HID1	1009	User: -/- Supervisor: R/-	Для просмотра конфигурации PLL (используется для "разгона" CPU)
HID2	920	User: -/- Supervisor: R/W	Для включения особых режимов Gekko (Paired-Single и управление блокировкой L1-кэша)
IABR	1010	User: -/- Supervisor: R/W	Instruction address breakpoint. Точка останова на исполнение инструкции.
ICTC	1019	User: -/- Supervisor: R/W	Управление выборкой инструкций.
L2CR	1017	User: -/- Supervisor: R/W	Управление кэшем второго уровня.
MMCR0,MMCR1	952,956	User: -/- Supervisor: R/W	Управление счетчиками производительности (performance counters)
PMC1-PMC4	953,954, 957,958	User: -/- Supervisor: R/W	Performance Counters
UMMCR0,UMMCR1	936,940	User: R/- Supervisor: R/-	MMCR0,MMCR1 для пользовательского режима.
UPMC1-UPMC4	937,938, 941,942	User: R/- Supervisor: R/-	Performance Counters для пользовательского режима.
SIA	955	User: -/- Supervisor: R/W	Содержит примерный адрес текущей инструкции. Т.к. у Gekko инструкции выполняются конвейером, поэтому точный адрес не всегда можно указать.

SDA	959	User: -/- Supervisor: R/W	Оставлен для совместимости с другими моделямми PPC.
USIA	939	User: R/- Supervisor: R/W	SIA для пользовательского режима.
USDA	943	User: R/- Supervisor: R/W	Оставлен для совместимости с другими моделямми PPC.
THRM1-THRM3	1020-1022	User: -/- Supervisor: R/W	Для управления питанием.
WPAR	921	User: -/- Supervisor: R/W	Для управления буфером подкачки (Write-gather buffer)

Дополнительные регистры, определенные архитектурой PowerPC, но которых по идее не должно быть у Gekko:

SPR	N	Доступ	Описание
PIR	1023	User: -/- Supervisor: R/W	ID процессора (идентификационный номер). Используется в мультипроцессорных системах.

Названия некторых регистров звучат "слэнгово", видимо разработчики архитектуры использовали их в повседневной речи, а в последствии так и оставили названия. Например регистр DEC - Decrementer, дословно вообще нельзя перевести на русский язык, так как слово decrement (уменьшить значение чего-либо на 1) не имеет аналога. Регистр XER вообще никак не расшифровывается, зато по русски дословно звучит прикольно :)

Смысл теста заключается в "подтверждении" официальной документации. Иногда бывает так, что в ней привираются кое-какие факты, или допускаются обычные опечатки. Программа для тестирования несложная. Мы создаем свой обработчик программного исключения Illegal Instruction. Вначале читаем и записываем SPR в режиме супервизора (SPR 0...1023), а потом переключаемся в режим пользователя и повторяем операцию. Для определенных регистров мы используем "магические" значения, для остальных - значение 0xFFFFFFFF. Обработчик исключения устроен специальным образом так, что при неверном доступе к SPR он выводит информацию об исключении и подвешивает Gekko. Результаты теста выводятся на экран. Во время тестов могут возникнуть всего два исключения: Program Privileged, когда доступ к супервизорскому SPR производится в режиме пользователя, и Program Illegal, когда производится доступ к отсутствующему или Read/Write-Only SPR

Результаты тестирования.

[Обсудить в гостевой]

---

6 Апреля 2007. Краткое описание Gekko (PowerPC).
Центральный процессор GC - Gekko основан на архитектуре PowerPC G3 (третье поколение), источники в интернете указывают на модель 750CXE, но это не особо важно, так как Gekko абсолютно совместим с 32-битной "Generic" архитектурой PowerPC. Вообще в архитектуре PowerPC модель процессора не имеет особого значения, так как в основном она влияет на производительность. Если программа написана на Generic (дословно означает "общая") PowerPC, то она гарантированно будет работать на всех моделях. Но всё-же модели отличаются друг от друга дополнительными "фичами", использование которых нацелено на увеличение производительности программ.

Вот список того, что было добавлено "эксклюзивно" в Gekko:
- Набор инструкций Paired-Single;
- Кэш может работать в режиме блокировки как scratch-pad буфер. Обмен между залоченным кэшем может происходить через DMA или прямой записью.
- Performance Counters. Счетчики для анализа производительности программ. С помощью них программист может узнать количество выполненных инструкций, попаданий/промахов в кэш и много чего ещё;
- Кэш второго уровня (256 KB);
- Блок предсказания переходов и таблица памяти переходов, для оптимизации циклов. (Знакомым с ассемблером PowerPC пригодятся суффиксы "+" и "-" в инструкциях перехода);
- Write Gather Pipe (буфер подкачки); Довольно часто используется для быстрой передачи графических данных (пакетов);
- Управление питанием. Доступно три режима: DOZE, NAP и SLEEP. Хотя эти режимы не используется в играх/программах и Dolphin SDK;
- Отладочный интерфейс для исполнения программ по шагам и поддержка аппаратных точек останова;

Теперь краткое описание того, что доступно программисту PowerPC:
- В процессоре есть 32 целочисленных 32-разрядных регистра (обозначаются r0-r31) и 32 вещественных 64-разрядных регистра (fr0-fr31). Над целыми числами можно применять операции сложения, вычитания, умножения, деления, арифметического/логического сдвига, а также логические операции (AND, OR, итп). То же самое можно делать с

вещественными числами формата IEEE-754 одинарной или двойной точности (+, -, *, /), в добавок есть также операция вычисления квадратного корня SQRT и вычисление полинома A * B + C. Отдельно нужно сказать о такой уникальной целочисленной операции как RLWINM - расшифровывается как "Сдвинуть циклически операнд влево на n бит, а потом применить к нему операцию AND)", математически можно записать как R = ROTL(A, n) & MASK. Это черезвычайно мощная и универсальная операция широко используется компилятором для оптимизации таких блоков языка Си, как например: if((a >> 8) & 0xFF) - это выражение будет откомпилировано как одна(!) инструкция RLWINM (включая проверку if).
- Режимов адресации памяти всего два: регистр + регистр и регистр + смещение. Джентельменский набор.
- Процессор работает в двух режимах: пользователя и супервизора. Различие в том, что некоторые сугубо системные инструкции и области памяти не могут быть выполнены в режиме пользователя. Короче говоря тут можно провести аналогию с реальным и защищенным режимом X86.
- Доступ ко всем системным функциям процессора осуществляется через Special-Purpose Registers ("регистры специального назначения").
- Для сравнения чисел используется специальный регистр - CR (Condition Register). Он содержит 8 полей, в каждом из которых находятся флаги результата сравнения. Непосредственно после операции сравнения идет инструкция перехода, которая анализирует состояние выбранного поля регистра CR и совершает (или не совершает) переход. Ещё есть такая удобная вещь, как сравнение результата текущей операции с нулем. Для этого в ассемблере используется суффикс точка ("."), например выполнение инструкции add. r3, r4, r5 будет проходить следующим образом: сложить r4 и r5, поместить результат в r3 и сравнить результат с нулем. Результат сравнения поместить в поле CR[0].
- У PowerPC нет инструкций "Jump". Все переходы осуществляются инструкциями "Branch" (досл. ветка), коих великое множество. Вызов процедур реализуется через специальный регистр - LR (Link Register). Чтобы вызвать процедуру нужно выполнить инструкцию "Branch And Link", которая сохранит адрес возврата в регистре LR. Аналогом "Return" является инструкция blr - "Branch to Link Register". Циклы типа FOR I=1 TO N реализуются с помощью регистра CTR (Counter). Специальная инструкция перехода уменьшает CTR на 1 и совершает переход в соответсвии с условием (CTR равно/не равно 0).

Размер инструкции составляет 32 бита. Дизассемблированный код PowerPC выглядит примерно так:

```
8135D8A8  80A10008  lwz         r5, 8 (r1)
8135D8AC  8101000C  lwz         r8, 12 (r1)
8135D8B0  54A6007E  rlwinm      r6, r5, 0, 1, 31
8135D8B4  7C060000  cmpw        r6, r0
8135D8B8  90830000  stw         r4, 0 (r3)
8135D8BC  38E50000  addi        r7, r5, 0
8135D8C0  38860000  addi        r4, r6, 0
8135D8C4  4080000C  bge-        0x8135D8D0
8135D8C8  7C804379  or.         r0, r4, r8
8135D8CC  4082000C  bne-        0x8135D8D8
```

Чтобы узнать производительность Gekko относительно Intel/AMD я провел небольшой тест - умножение большого количества матриц. Тест получился очень умозрительный и по результатам Gekko примерно равен 1000 MHz Pentium (это с учетом оптимизаций PairedSingle/SSE). Ну то есть вычислительные мощности куба примерно были равны компьютерным на момент выпуска консоли.

Ресурсы по статье.
- Руководство по программированию PowerPC. (Довольно туманный документ, который можно сократить до 20 страниц. Но содержит описание работы инструкций, так что бывает нужен)
- Обзор архитектуры PowerPC (MS Word). (Сокращенный вариант руководства, русский перевод)
- PowerPC на Wikipedia. (В основном рассказывается об историческом развитии архитектуры и какие есть модели процессоров)
- Дизассемблер для Gekko (исходный код).
- Ещё один дизассемблер для Gekko (Более усовершенствованная версия. Также поддерживается архитектура POWERPC-64)
- Полный список инструкций Gekko.

[Обсудить в гостевой]

---

11 Мая 2007. Загрузка Gamecube.
Информация о том как загружается куб долгое время была почти мифом, потому что инженеры Nintendo решили сделать крипто-защиту на BootROM. Но всё равно, спустя некторое время хакеры нашли способ обходить эту защиту и задампили бутром куба.

Внешний вид и расположение чипа Macronix с BootROM. Рядом показан ASIC FLIPPER.

Аппаратно BootROM представляет собой небольшую микросхемку Macronix. Сам ROM состоит из 2 банков, общим размером 2 MB. В состав этой микросхемы также входят часы реального времени (RTC) и память на батарейке (SRAM). Доступ к данным осуществляется 32 байтовыми пакетами через DMA-канал, по шине EXI. Дополнительно первый банк (1 MB) доступен по адресам 0xFFF00000...0xFFFFFFFF. Сделано это для того, чтобы процессор мог выполнить программу начальной загрузки (адрес RESET у Gekko сделан 0xFFF00100).

Крипто-защита основана на алгоритме псевдослучайной генерации чисел через LFSR [1] и операцией XOR между PRND-числом и данными. Причем эта операция производится "на лету" во время DMA-пересылки. Конечно, такая защита не надежна, потому что данные передаются по шине уже в расшифрованном виде. Именно так и сдампили бутром (подсоединившись к шине), после чего стало возможным получить поток XOR-ключей, произведя операцию XOR между зашифрованными и расшифрованными данными. Нужно отметить, что Nintendo сделала выводы, и у Wii бутром располагается внутри процессора, так что сдампить простыми методами его уже нельзя.

Теперь следует рассказать, что происходит после того как на кубике нажата кнопка POWER. После нажатия POWER, на микросхемы Gekko и Flipper подается сигнал аппаратного сброса. Что делается внутри Flipper-а точно не известно, так как это закрытая информация, но очевидно, что там очищаются все регистры и внутренняя видео-память. Что происходит в Gekko известно очень хорошо, это написано в "PowerPC Programming Manual": выполнение программы начинается с адреса 0x00000100 или 0xFFF00100. Какой именно адрес использовать определяется битом MSR[IP]. После аппаратного сброса, у Gekko этот бит=1, соответственно адрес программы начальной загрузки будет 0xFFF00100. Программа начальной загрузки у куба называется **BS** (наверно по аналогии с Bootrstrap Stage в UNIX). Занимает BS всего пару килобайт, и делает следующее:

- Инициализировать Flipper (в частности сбросить DVD)
- Инициализировать Gekko (включить кэширование, трансляцию адреса, настроить MMU).
- Проверить память 1T-SRAM (в память записываются определенные значения, и если после прочтения они не такие как надо, то значит память повреждена).
- Загрузить вторую часть (графическую оболочку) и запретить чтение из бутрома.

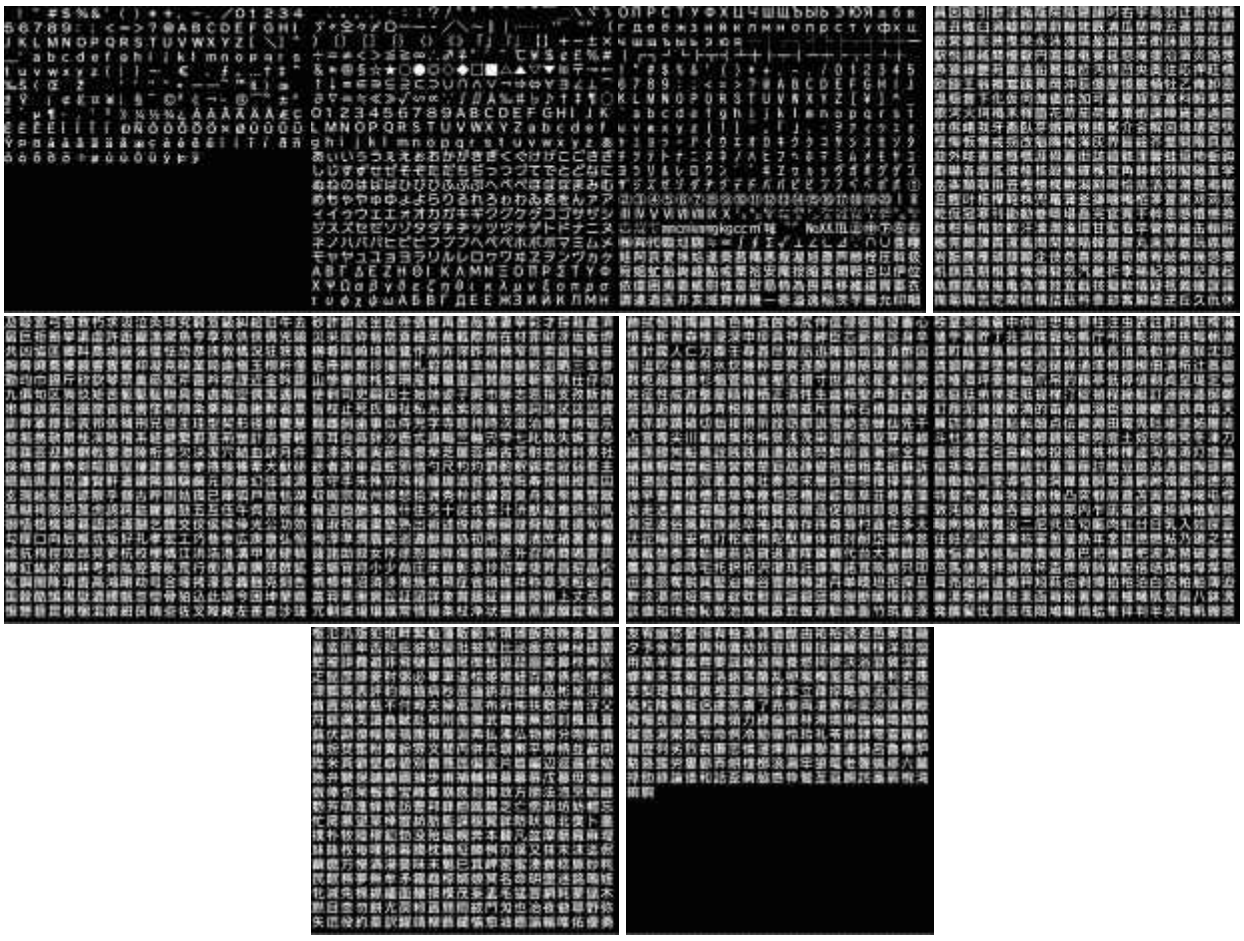
Вторая часть это не что иное, как графическая оболочка. Официальный термин, который для нее использует Nintendo - **BS2** или **IPL** (Initial Program Loader). Размер оболочки довольно увесистый - 1.5 MB. Программно в IPL нет ничего особенного, можно сказать что это обычная DOL-программа, зашитая внутри бутрома. У Gamecube нет централизованной ОС (Dolphin OS - это статическая библиотека, которая линкуется с DOL-программой), и IPL не исключение. В его состав входит старая (видимо самая первая) версия Dolphin OS.

Внешний вид графической оболочки. Выглядит как вращающийся стеклянный кубик, внутри которого плавают замысловатые фигуры.

Финальной стадией загрузки игры является **Apploader** - специальная программа, которая находится на DVD. Она помогает IPL загрузить главную DOL-программу. Именно Apploader указывает IPL адреса секций кода и данных для загрузки в память. В теории формат исполняемого файла может быть любым, но на практике Apploader-ы Nintendo понимают только исполняемые файлы формата DOL. Работает Apploader довольно просто, у него есть специальный вызов, который возвращает 3 параметра: адрес памяти, смещение на диске и размер данных. IPL получив эту информацию загружает данные с DVD до тех пор, пока Apploader возвращает параметры. Затем IPL переходит на точку входа в DOL-программу.

Важное замечание: IPL и Apploader загружаются в память по фиксированным адресам - 0x81300000 и 0x81200000 соответственно. Поэтому в DOL-программах запрещается создавать сегменты в районе этих адресов, чтобы не затереть программу загрузки.

Ещё в бутроме есть куча шрифтов, которые сжаты оригинальным алгоритмом, очень похожим на LZ:



Набор ROM-шрифтов.

Эти шрифты используются графической оболочкой IPL, и иногда играми (например в Bust-a-Move 3000). Хотя как правило разработчики игр делают свои собственные шрифты. При чтении шрифтов бутром не использует шифрование, так что их можно использовать и после загрузки IPL. Выложить оригинальные шрифты я не могу, но в эмуляторе Dolwin 0.10 есть модифицированные шрифты, максимально похожие на оригинальные. Не знаю кому они могут понадобиться, но работа была сделана немаленькая, попробуйте переделать столько закорючек сами!

Как заключение, вот вкратце последовательность загрузки куба:
- **BS**, маленькая программа которая загружает IPL, и выключает дешифратор, чтобы нельзя было сдампить бутром;
- **IPL**, графическая оболочка, где можно настроить опции, дату/время и полазить по карте памяти;
- Если в привод вставлен DVD, то вместо графической оболочки, с помощью **Apploader** загружается игра.

[1] LFSR (Linear Feedback Shift Regiter, Регистр сдвига с обратной связью) широко используется в схемотехнике для генерации псевдослучайных чисел. Устроен LFSR как обычный регистр сдвига, но выходной разряд подается назад на входные каскады, где стоят вентили XOR. Несколько LFSR также можно объединять в цепочки, для получения требуемого закона распределения псевдослучайных чисел.

Ресурсы по статье. [Вы **не** найдете тут каких-либо ромок! Всё законно.]
- Разбор работы бутрома начиная с вектора RESET.
- Разбор работы стандартного Apploader.
- Исходный код BS.
- Описание алгоритма сжатия шрифтов.
- Программа для запаковки/распаковки шрифтов (исходный код).

[Обсудить в гостевой]
