Аппаратные и программные аспекты видеопроцессора V9990

RomanichApparate@mail.ru

Введение

Современная электроника набирает такие бешеные темпы развития, что успеть уследить за всем, а тем более использовать не удаётся! Появление ASIC'ов потрясло электронику (да и радиотехнику как приложение тоже). Ещё вчера я лично, не смел мечтать и думать о том, что знаю и умею сейчас! Развитие напоминает движение по экспоненциальной спирали – чем дальше, тем быстрее и больше… Иногда просто кажется, что не хватит жизни одного человека *успеть* разобраться со всеми ноу-хау ASIC'ов! Кто не знает, поясню, ASIC – это Application Specific Integral Circuit – интегральная схема специализированного назначения. Это могут быть чипы MPEG-кодека, музыкальный сопроцессор, графический ускоритель и многое другое… Появление ASIC'ов обязано достижениям (V)LSI – (Very) Large Scale Integration – высокой степени интеграции дискретных элементов на единицу площади кристалла чипа.

К сожалению, в мире сложилась такая ситуация, что последние новшества техники (в частности новейшие чипы) не документированы с аппаратной и программной точек зрения! Составлены только лишь описания по использованию готовых устройств, собранных на базе этих чипов. Конечно, крупным корпорациям, заключившим договор и платящим немалые денежки конторам, производящим ASIC'и вся документация доступна – как железячникам, так и программистам. А что делать обычным любителям или полупрофессионалам, не имеющим больших денег, но жаждущим творчества? И не имеющим порой даже юридического статуса, открывающего доступ к новым ASIC'ам. Ответ один и он очевиден – *ждать*! Как говорится, время лечит…

Ну вот, подобным образом компания Yamaha в 1991-м году выпустила на свет чудо того времени – видеоконтроллер (с ускорителем) V9990. Естественно всё было так, как написано в предыдущем абзаце… Сегодня же 2007-й год, и некоторые вещи просто морально и физически устарели. Но статуса ASIC'а V9990 не лишился! И видеочип V9990 практически полностью открыли (как по части железа, так по части и софта). И это радует – сработал вывод из предыдущего абзаца! И в наше время есть поклонники этого видеопроцессора. Несмотря на его древность, его возможности меня просто впечатляют!

Я не буду подробно рассказывать о возможностях этого видеочипа – в Интернете навалом инфы про него. Вкратце скажу, что это – гибрид двух видеорежимов – спрайтового и битово-картового. Первый применяется в игровых консолях недавнего прошлого времени, второй практически совпадает с видеорежимами IBM PC-совместимых машин. Желающие могут поискать в Интернете об этом поподробнее… Вместо этого, я расскажу, как можно программировать V9990, который (несмотря на количество выводов 128 с шагом 0.8мм) установлен в отладочную плату (собрана мной "на коленках"!) под названием "Video Monster".

Следует отметить, начальные эксперименты с V9990 мной скрывались от "чужого взора" (весна-лето 2006 г.) с целью, чтоб никто не мешал работать и накопить достаточно опыта. Да и грязь чтоб никто не лил, в процессе освоения V9990. Далее я, был включён в массовую рассылку несколькими лицами из NedoPC Group. Участники рассылки помогли мне решить несколько зловредных проблем, связанных с V9990, но о них речь впереди. Затем рассылка канула в небытие… В основном всё "подводные камни" (особенно по части железа) были найдены и ликвидированы (или их обошли)! Далее на zx.pk.ru я объявил о своём достижении и показал возможности этого графического чипа – лагерь поклонников Спектрума разделился – одни были "за" использование V9990 как дополнительного видеоускорителя в Спектруме, другие "против" – но это тема уже *другого* разговора. Жаждущие могут почитать топик "VideoMonster на V9990" на zx.pk.ru

В поисках идеальной видеосистемы

Всё началось с того, что после сборки своей МикроМашины, я увидел недостатки LCD (ЖК-дисплея). Серьёзные недостатки таковы:

1) Отсутствие стробоскопичности (one-frame эффекта) как на ЭЛТ
2) Большая инерционность LCD-матрицы
3) Большинство GSM-LCD рассчитаны на шину SPI (последовательная передача данных), вместо параллельной шины, которая обладает лучшей пропускной способностью – что становится главным критерием при передаче данных из памяти в контроллер LCD
4) Невысокие яркость и контрастность
5) Очень малый угол обозрения

Все вышесказанные недостатки касаются большинства STN GSM-LCD, конкретно GSM Nokia 7210 LCD, в состав которого входит контроллер от Epson S1D15G10, поддерживающий автономно регенерацию изображения и некоторые примитивы 2D-ускорителя. Его максимальная пропускная способность до 20Мбит/с (я насиловал его до 24 Мбит/с).

Поэтому после создания МикроМашины (буквально едва не написав ещё софт под неё) появилось желание построить новую ~Машину, с устранёнными недостатками (и не только по видео-части)! Всё упиралось в видеосистему. Рыскание по Интернету ни к чему хорошему не приводило – то, что хотелось, было просто *нереально* заказать! То чип недоставаем, то нужен юр-статус, то нет документации, то нет огромных денег, то ещё чего…

Ситуация круто изменилась, когда в один прекрасный момент на том же zx.pk.ru нашёл интересный топик (и даже не один!), в котором велись дискуссии по применению V9990 в качестве видео-ускорителя для Спектрума. Далеко и тяжело ходить не пришлось – в наше время чип полностью документирован и самое главное – *доставаем*! К тому времени, я уже был знаком (пока, к сожалению, только заочно) с Чуниным Романом (CHRV), который как раз и смог достать несколько образцов V9990 и главное – он имел желание рассылать их поштучно заинтересованным людям (в том числе и мне) по почте!

Ну вот, пришло время – пришла мне пара микросхем V9990 и память к ним. Осталось дело за малым – разобраться с этим видеочипом и сделать “на коленках” отладочную на нём. Следует отметить, что информации из даташита на V9990 может не хватить на то, чтоб узнать, каким же образом цеплять память на видеочип, чтоб всё нормально заработало с первого раза без “наступания на грабли“. Поэтому CHRV помог мне с этим разобраться – дал мне принципиальную схему GFX9000, использовавшегося в старших версиях компов MSX.

А тут ещё время отпуска подходило (август месяц 2006 г.). Взяв отпуск на работе, я собрал необходимые вещи (IBM PC, элементную базу, паяльник ит.п.) и уехал из суетливого Владивостока в Тихоокеанский (где провёл большую часть детства) к родителям.

Вот в это время-то и попёрли *первые* эксперименты с V9990. Времени было куча (в отпуске всё же!) – не только химичил с V9990, но и был на пляжах, даче и пр.

Первые шаги

В это время я преодолел страх перед невозможностью изготовления методом ЛУТ (лазерно-утюжная технология изготовления печатных плат) дорожек толщиной 0.3 мм. Использовал всё самое свежее – новый картридж с тонером для Ч/Б лазерного принтера, тонкую глянцевую бумагу и главное – свежее хлорное железо, разведённое в тёплой воде (+40..+50ºC). На удивление плата вытравилась (за полчаса) успешно – даже порывов небыло! Следующие два дня потратил на пайку компонентов к плате. Чуть раньше был написан первый софт для управления V9990. Для управления использовался LPT-порт (был очень рад, когда узнал, что его можно в ECP-режим загнать – для *чтения* данных помимо записи). Программа писалась на Turbo Pascal 7.0, но легко портировалась на TMT Pascal DOS Edition и работоспособна под DOS и Win9x.

В связи с тем, что ножек LPT-порта мне не хватало, я микросхемой 74HCT573 удвоил часть ножек, потратив одну (тоесть, по сути, реализовал свёртку шин адреса и данных для V9990), но об этом ниже.

Помню, как сильно переживал перед *первым* включением отладочной платы на V9990 (Video Monster)! Использовался дешевенький TV Akai, комп P4 2.7GHz (LPT-порт которого мог сгореть в два счёта, особенно когда шина данных на *вывод* со стороны LPT и V9990). И первая программа вовсе не выводила изображение на экран TV, а… тестировала всю видеопамять V9990! Программа через LPT записывала байт 0x55 и 0xAA в каждый адрес VRAM V9990 и затем читала его – чтение было корректным!!! Ура! Значит, память подсоединена верно. Дальше – больше! Вогнал в VRAM rar-архив, затем считал его и успешно распаковал его… Всё это говорило в пользу того, что чтение/запись из/в видеопамять(и) через регистры V9990 происходит корректно.

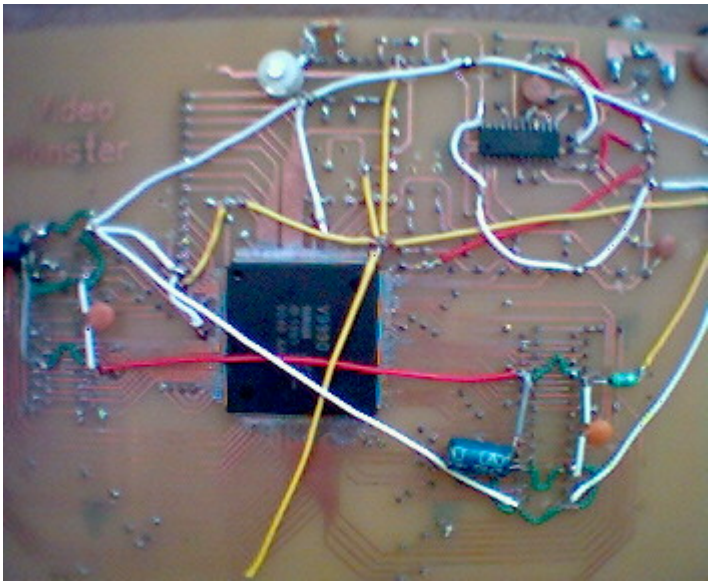
Захотелось дальше вывести что-то осмысленное на TV. Для упрощения вместо спрайтового режима использовался привычный всем PC-шникам бит-картовый режим. Но тут поджидала *первая* неприятность, связанная с отсутствием у меня кварца *ровно* на 21.47727 MHz. Первоначально у меня стоял кварц ровно на 20 MHz. Тут следует отметить, что V9990 требует два тактовых сигнала – это 14.31818 MHz и 21.47727 MHz. Первый кварц легко доставаем – из PC-шных материнских плат, видео/звуко-карт – в общем такие кварцы у меня, слава Богу, были. Ну второй же блин, целая проблема! Фактически – это частота первого, умноженная на 1.5 (умножить на 3, поделить на 2). Вот таких кварцев-то и ни у кого не было!!! Со злости я уже катил на Ямаху бочку, что PLL не организовали внутри V9990 или взяли бы более доступный кварц! Или одним первым кварцем обошлись – замечательно было бы всем! Ещё как назло, я выбрал видеорежим, использующий 20 MHz и очень долго ругался из-за того, что только при первом включении VideoMonstr'a появляется картинка на экране, при последующем программировании вся TV развёртка срывается – горизонтальные полоски кадра начинают перекручиваться, образуя вместо нормального изображения какой-то ливер. Ещё не понравилось то, что картинка была чёрно-белая. Как я уже сказал, использовался дешёвенький китайский телевизор Akai, затем я попробовал на более дорогом телевизоре (HITACHI) – он вообще никакого ливера не выводил – просто культурно посылал видеоизображение на три весёлых буквы – высвечивал синий экран с надписью "No Signal". Стоило подержать VideoMonster отключенным от питания на минут 5-10, затем снова включаем – картинка есть. Но при последующем программировании опять ливер на дешёвом телевизоре…

Подводные камни V9990

Забив на режимы, требующие такты 21.47727 MHz, решил попытать счастье в бит-картовых режимах, тактирующихся от 14.31818 MHz. Следует отметить, что V9990 не имеет встроенного осциллятора (a.k.a. XTAL) для привешивания чисто кварца. Пришлось собирать генератор на 74HC02. И надо сказать, что после инициализации этого видеорежима (384x290 OVERSCAN PAL) наконец-то удалось получить устойчивое изображение! Но оно было чёрно-белым… Это уже второй подводный камень. Долго копать не пришлось – в то время я использовал кодер-PAL CXA1645, который не имел XTAL'а – для него тоже пришлось собирать отдельный генератор поднесущей PAL на 4.433618 MHz. Такой кварц, к счастью, тоже распространён, только вот беда – у каждого кварца есть паразитная ёмкость, которая мешает реализовать частоту поднесущей PAL с точностью до 50Hz! И везде в даташитах на кодеры приводят схему внешнего генератора поднесущей PAL с *подстроечным* конденсатором. Далее поставил себе тоже подстроечник в генератор, покрутил и… картинка стала в цвете!!! Только вот крутить осторожно надо – всё-таки точность 50Hz – это высокие требования на единицах МГц. Вот одна из первых выведенных картинок:

Казалось бы – вроде ничего, но если присмотреться повнимательнее – видно множество вертикальных полосок с небольшим искажением цветов. Вот вам подводный камень номер три! Надо сказать, что это один из самых вредоносных камней. И избавиться малоопытным аппаратчиками без достаточного опыта в *разводке* печатных плат сразу не удастся! Проинформировав CHRV о своих достижениях и проблемах, он немедленно подключил меня к массовой рассылке, где обитали люди, заинтересованные в V9990 (NedoPC Group). Так как CHRV был сильно занят, то на помощь пришёл другой человек из группы – Виктор Рощупко (Ronin). Вот он и помог решить эту проблему. Впервые от него я узнал, что цепи земли, питания и сигнальные – нужно разводить в отдельности. Проще говоря – земляной контур и контур питания в цифровой части и аналоговой должны быть разведены *отдельно* и соединяться только в *одной* точке (желательно)! Цифровые сигнальные линии (дорожки видеопамяти, порты V9990) не должны проходить близко параллельно аналоговым сигнальным линиям (R, G, B, CSync)!!! Результат – наводки и импульсные помехи, выражающиеся в виде картинки выше. И ещё одна неприятность, о которой рассказано будет чуть-позже.

Пришлось превратить печатную плату в утиль-сырьё. Отодрал ножом все дорожки питания и земли – проводами сделал новые контуры раздельных питаний и земель. В итоге плата стала иметь такой плачевный вид:



Но зато результат изменился круто в нужную сторону! Вертикальные полоски исчезли! Картинка стала чистой (не хуже чем на мониторах IBM PC):

Далее я решил вернуться к режимам, тактирующихся кварцем на 21.47727 MHz. Тщательно у себя поискав, нашёл кварц на 21.15 MHz и поставил его вместо 20 MHz. И режимы пошли!!! Правда экран немного подёргивался. И тут я заметил любопытную вещь – захват цвета зависит не только от точности частоты генератора поднесущей PAL, но и от точности тактовой частоты! Чем больше будет отклонение от 21.47727 MHz, тем более будет неустойчива цветность, а диапазон подстроечного конденсатора сузится. И всё-таки мне удалось настроить поднесущую так, чтобы цветность держалась даже с “чуть неправильным” кварцем!

Далее решил заняться освоением блиттера V9990. По сути – это 2D-ускоритель, аппаратный интерфейс которого напоминает PipeLine Voodoo2, 3. Но не тут-то было – вылез камень номер четыре! Его суть заключается в том, что когда активно используются

команды блиттера (рисование прямоугольников, блиттинг и др.), то происходит МЕГА-глюк – содержимое видеопамяти подсирается мусором. Возможны офигенные подёргивания отдельных групп строк на экране TV. Долго я боролся с этим – ставил ёмкости побольше – электролиты, керамику… Даже дроссель по питанию персонально к *каждой* микросхеме видеопамяти! Удалось избавиться опять-же переразводкой контуров питания и земли – только уже для видеопамяти отдельно. Хотя этот вопрос остался не закрытым – видать переходные процессы, происходящие в чипах VRAM, вызывают мощные наводки на сами чипы… Следует отметить, что если все четыре микросхемы VRAM посадить на L7805CV, то он будет очень тёплым, что говорит прожорливости чипов памяти!

И наконец, камень номер пять. Называется “бегущие переливы”, возникает на мой взгляд, из-за того, что уж слишком много опорных частот требует V9990, а главное – частота поднесущей PAL и тактовая частота колеблются не синхронно (есть фазовый сдвиг). Проще говоря – это “две разных частоты”. И поэтому этот эффект особо виден на контурах объектов изображения – контуры как-бы немножко изгибаются – закручиваются как молекулы ДНК! Избавиться от этого можно применив один *единственный* опорный генератор в схеме. Например, какой-нибудь PLL с программной перестройкой частоты, имеющий более одного выхода для продуцируемых частот. Или ещё проще – перейти на NTSC-стандарт вместо PAL. В этом случае вопрос решается автоматически – у V9990 есть ножка NTSC-subcarrier сигнала (я думаю, что она возникает внутри V9990 путём деления 14.31818 MHz на 4 и/или 21.47727 MHz на 6) – колебания будут синхронны и от одного источника. Данный “косметический недостаток” не особо бросается в глаза, поэтому при исполнении схемы в режиме PAL, на него можно забить!

Теперь следует предостеречь от ошибки, которая способна вывести из строя V9990. При выполнении внешних генераторов на CMOS-микросхемах (74HCxx) ни в коем случае нельзя выходы вентилей подсоединять напрямую к входу MCKIN V9990! Видеочип разогреется как подошва утюга (буквально за десятки секунд)!!! Это не шутка – я вовремя заметил, выключил питание, дал видеочипу остыть. Затем подключил выход вентиля через керамический конденсатор 0.1мкФ. Хорошо, что не успел сжечь V9990…

Ну вот, все обнаруженные мной хардварные камни V9990. Очень буду рад, если кто-нибудь сообщит о неизведанном… Далее поговорим поподробнее о видеорежимах V9990.

Sprite Mode vs. Bitmap Mode

Конечно, первыми я задействовал бит-картовые видеорежимы. Человеку, который программировал железо IBM PC в прошлом, легко понять эти режимы. Подробно описывать, что такое Bitmap Mode и 2D-engine я не буду – прочитайте книги про акселераторы видеокарт и о Direct Draw в целом. Под Direct Draw не следует понимать только мелкософтовское творение – это общее название 2D-ускорителей, основные признаки которого:

1) Bitmap Mode
2) Наличие буферизации (передний буфер (отображение), задний буфер (рендеринг), вспомогательный буфер (спрайты (не тайлы!)))
3) Наличие команд блиттинга, флиппинга и других 2D-фич

Чем же хороши Bitmap Mode у V9990? Тем, что прежде всего – большая глубина цвета. Одновременно может отображаться 32768 цветов, 256 цветов. Я молчу уже о 16 и 2 цветах. И все команды 2D-ускорителя производятся записью в десяток регистров V9990. Довольно всё просто и красиво! Разработчики V9990 даже продумали такие моменты как цвет прозрачности, растровые операции и битовые маски, прокрутка изображения… Но блиттер меня очень быстро разочаровал из-за своей крайне *низкой* скорости рендеринга! Сразу скажу – написать реактивный Direct Draw не получится, всё-таки 14/21 МГц мало… Для написания игр с этим режимом следует расстаться, к сожалению. И попробовать Sprite Mode.

Спрайтовый (тайловый) режим у V9990 аналогичен видеорежимам игровых консолей прошлого века. Скажу, что режим очень *быстрый*, но программировать его на порядок сложнее – трудно привыкнуть к организации видеопамяти, нужно иметь тайловый склад ума (как у япошек, например)! Желающих просветить себя по тайловым режимам я могу послать к техническим описаниям видеосистемы упомянутых игровых видео-приставок (NES,SEGA MD, SNES). Лучшая, на мой взгляд, документация – это SegaTech, переведённый Игорем Внуковым (HardwareMan). Кроме того, глубина цвета в Sprite Mode слишком маленькая – одновременно 16 на тайл (как в SEGA MD). Это обстоятельство меня бесит сильнее, чем сложность программирования. Могли Ямаховцы сделать побольше (хватит 256 как в SNES)? Могли, да не захотели…

Тем не менее, за большее количество цветов можно побороться – разбить спрайт на несколько тайлов, окрашенных цветами из разных палитр (их четыре, нулевой цвет прозрачный). Итого имеем уже чистых 60 цветов. Кроме того, можно вытворять “фокусы” со строчной развёрткой (прерывание по строкам) и каждый раз менять палитру. Мне удавалось удвоить количество цветов – спрайт при попадании в верхнюю часть экрана, окрашивался одной группой цветов, при попадании в нижнюю часть – другой группой. Но и тут ждёт неприятный сюрприз – возникновение “снега” при смене палитры, причём замечено, что снег не появляется, когда трогаешь только нулевой цвет (фон/прозрачность). Поэтому для написания крутых игр под V9990, используя Sprite Mode, придется извращаться, причём выворачиваясь наизнанку! Другая трудность заключается в определении столкновения объектов – хочется ведь по-каёмке анализировать, а объекты не квадратные! Геометрия для точных столкновений не прокатит! В общем в целом можно сказать, что V9990 чем-то хуже, а чем-то лучше видеопроцессора SEGA MD. Тем не менее, вышесказанное не означает того, что игры писать нельзя под V9990, пример тому – бесчисленное множество игр под игровые приставки прошлого столетия, среди которых есть подлинные шедевры! Остаётся только белым цветом завидовать коллективам программистов, как умело они выворачивались на изнанку!

Аппаратная часть

Перейдём теперь к самому интересному. Файл VideoMonster.pdf содержит схему электрическую принципиальную VideoMonstr'a. Отмечу важные особенности схемы:

1) Вместо кодера CXA1645 применён CXA1145 (тот самый который в SEGA MD частенько стоит). Данный кодер хорош тем, что не требует внешнего генератора тактов для получения PAL subcarrier. Требуется всего лишь кварц (у меня на 4.4336 MHz). Если цветности не будет, подберите конденсатор, который включен последовательно с этим кварцем.

2) Для максимального упрощения схемы введено ручное переключение частот 14.31818 MHz/21.47727 MHz для видеорежимов V9990 с помощью джампера J1. Данный приём позволил избавиться ещё от одного тактового генератора – на ножку MCKIN приходит сигнал с XTAL! Не правда ли здорово? К сожалению, я не обладатель кварца на 21.47727 MHz – поставил то, что есть самое близкое 21.15 MHz (со сломанной УКВ радиостанции). Хотя возможны другие решения, но об этом чуть позже.

3) Джампер J2 разрешает использовать прерывания V9990. Очень полезная штука – особенно при подключении к микроконтроллеру (или процессору). Для LPT порта это бесполезно – он слишком медленный.

4) J3 разрешает использовать простои V9990. Они необходимы для шибко шустрых микроконтроллеров/процессоров, которые имеют бешеные циклы чтения/записи из/в регистров(ы) периферии (или память). Для LPT порта это ненужно.

5) Ножка MODE3 у V9990 занулена, так как это старший адресный бит, разрешающий обратиться к портам 8..15 видеочипа. Данные порты рулят канджой (которой нет) или reserved, поэтому не у нас не используются – поэтому разрешено обращаться только к портам 0..7.

6) V9990 не имеет чистого вывода выбора кристалла. Имеет только ~CSW и ~CSR, которые являются гибридами стробов записи/чтения с выбором кристалла. Поэтому если на шине висит более чем одно устройство, то необходим мини-дешифратор (74HC32 например), который из ~CSW и ~CSR делал бы ~CS, ~WR, ~RD.

7) Reset хардварный с помощью RC-цепочки. Желающие могут доработать, чтоб ресетить от контроллера. Хотя вряд-ли нужно – V9990 можно программно сбросить, установив/сбросив бит в нужном регистре.

8) Отнеситесь серьёзно к разводке земель и питания! Сигнальные линии делайте как можно короче. Ставьте насколько возможно близко кондёры (тантал+керамика) к микросхемам памяти.

9) Учтите, что выводы ~WAIT, ~INT0 и ~INT1 у V9990 с открытым стоком (или коллектором?) – в некоторых случаях понадобится pull-up резистор.

В заключение добавлю, что в этой схеме количество радиоэлементов минимизировано.

Файл VideoProgrammer.pdf содержит программатор для V9990 через LPT порт. Важные особенности схемы:

1) Резисторы по 100 Ohm нужны для защиты LPT порта и V9990 от перегрузок. При неправильной установке линий ввода/вывода возможен выход из строя как LPT, так и V9990 – если линии настроены на вывод с двух сторон одновременно! Будьте внимательны – не допускайте такой ситуации – резисторы могут не спасти! Сразу отмечу, что все мои программы (о них ниже) не допускают подобных случаев – поэтому можете смело их использовать.

2) Питание на программатор берётся от платы VideoMonstr'a. Светодиод на программаторе – монитор питания.

3) Регистр-защёлка 74HCT573 служит для "размножения" линий LPT порта – происходит расщепление на линии адреса и данных.

Файл VideoProgrammer.gif содержит разводку платы программатора. Синим цветом указаны перемычки – в случае односторонней платы.

Вот этого уже достаточно, чтобы начать экспериментировать с V9990!

А теперь немного поговорим о том, что можно изменить в лучшую сторону. Роману Чунину удалось добыть кварц на 42.954 MHz, то по сути дела в два раза больше, чем 21.47727 MHz. Напрашивается мысль о сборке делителя на 2 с помощью D-триггера или двоичного счётчика. Только понадобится уже 2 микросхемы – для делителя и для тактового генератора! Или кварц 42.954 MHz вешаем на XTAL V9990 и далее делим на 2 и подаём на MCKIN – одной микросхемой меньше, но требуется ручная коммутация кварцев. Или применить NTSC кварц на 3.579545 MHz и умножить его на 6 для получения 21.47727 MHz при помощи микросхемы ICS501 (хорошая, кстати, микросхема – PLL с параллельным управлением – никакого контроллера не нужно – джамперы помогут!). Лично я пока пробовал вариант с делителем на D-триггере (74HC74) плюс тактовый генератор (74HC00) – работает без проблем!

Кодер CXA1145 можно заменить на более современные CXA1645 или CXA2075 – обвеса у кодера в этом случае меньше, но требуется отдельный осциллятор на PAL поднесущую, или опять городить дополнительный генератор тактов!

Обсуждённый выше вариант показан в файле Fragment.jpg.

Программная часть

Все программы (с исходниками) лежат в папке V9990Soft. В ней три папки:

1) LPTport – программы для IBM PC (DOS, Win9x) – рулят V9990 через LPT порт.

2) AVRcontroller – программы для микроконтроллера Atmega8515. Регистры V9990 красиво вписаны в интерфейс внешней памяти контроллера!

3) Converter – всякие конвертилки BMP-файлов в форматы, нужные V9990.

Рассмотрим проекты LPTPort:

1) Bitmap Mode – демонстрация бит-картовых режимов (с блиттером!).

2) P1 Mode – демонстрация спрайтового режима.

3) VRAM Test – название говорит само за себя – тест всей видеопамяти.

Проекты AVRcontroller:

1) TestMachine – демонстрация спрайтового режима.

2) TestMachine2 – самое прикольное, что есть показать! Спрайтовый режим + прерывания (строчное и кадровое) + мультиколор фона (аж 4 цвета!). Спрайты передвигаются шустро и без глюков (синхронизируются по кадрам).

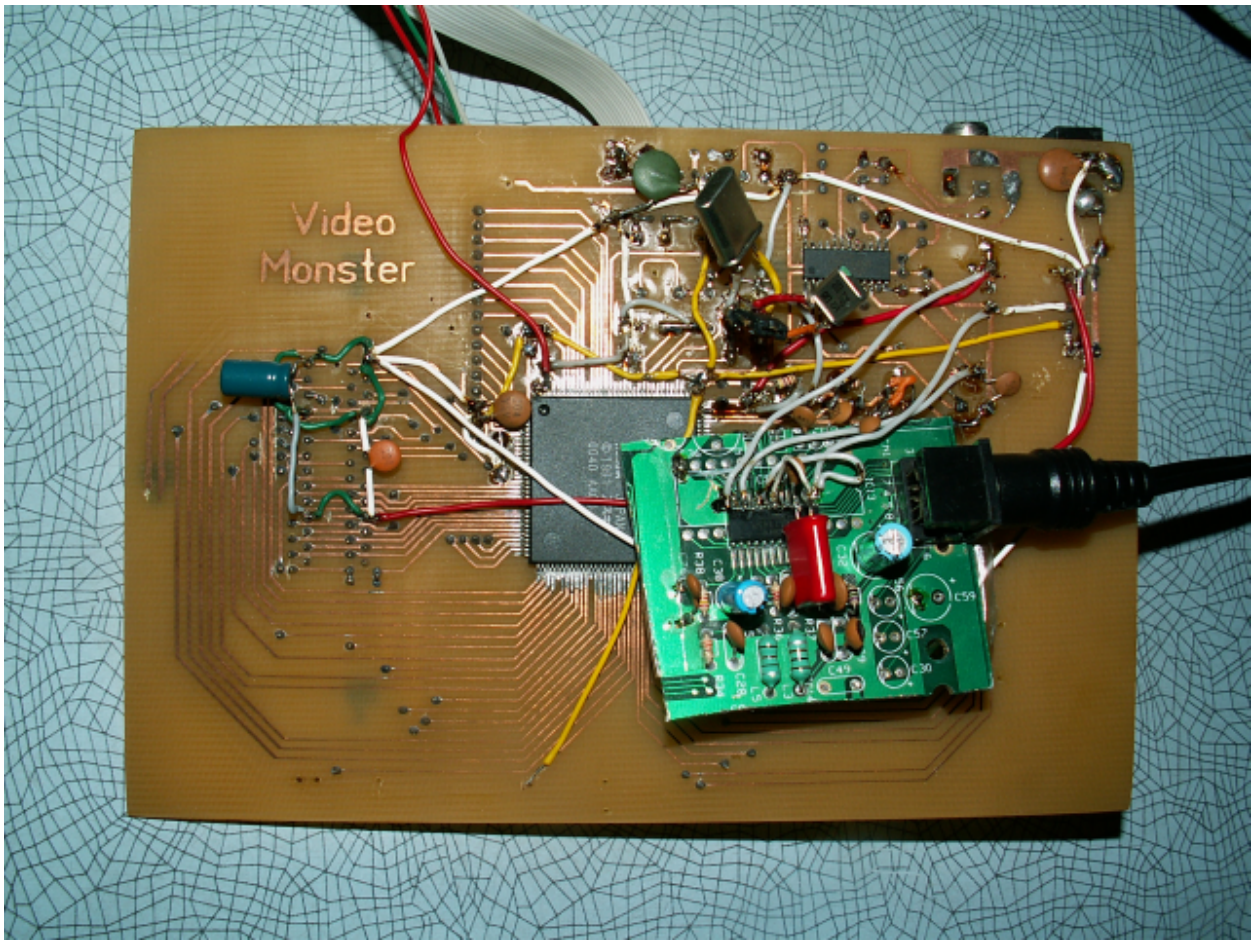
Необходимые среды программирования:

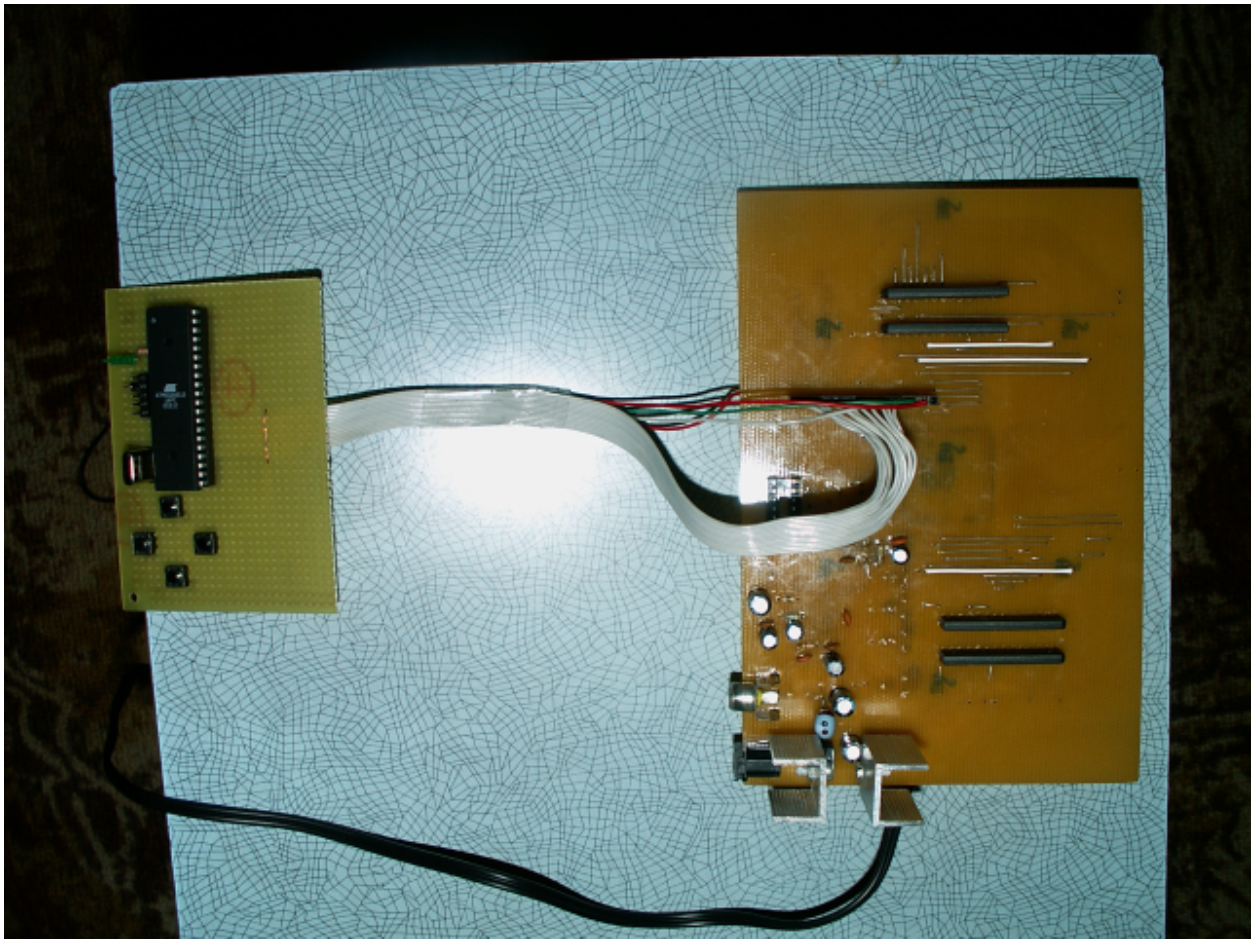
Turbo Pascal 7.0 или выше
TMT Pascal 3.90 DOS Edition
CodeVision AVR C 1.24.8b (может можно и помоложе).

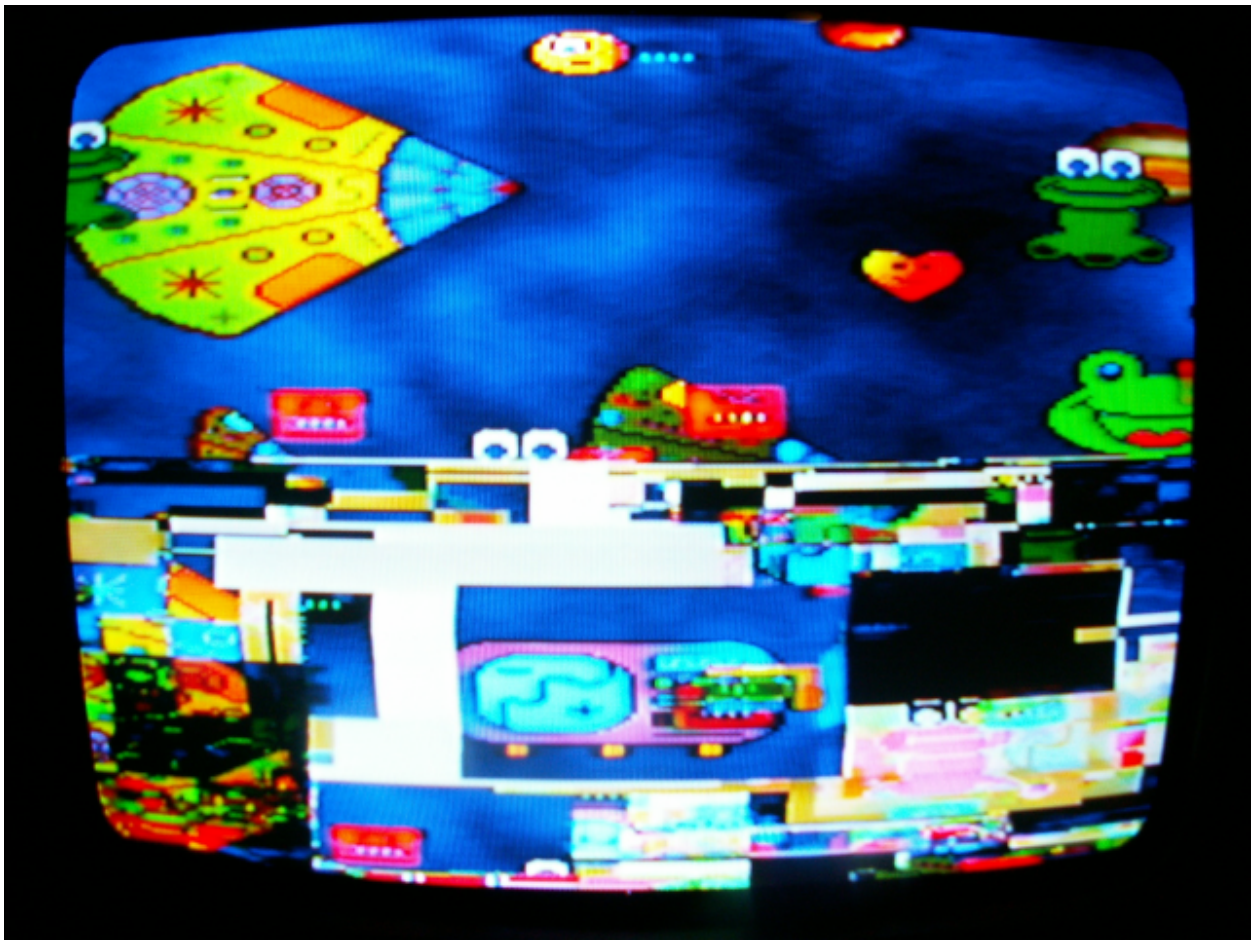
V9990 подключается к Atmega8515 через External SRAM Interface. Чтобы не ставить регистр-защёлку (см. даташит на Atmega8515) – адресные линии V9990 A0..A2 подключаются к A8…A10 Atmega8515. К ~INT0 Atmega8515 подключаются выводы ~INT0, ~INT1 V9990 – необходим подтягивающий резистор 10 kOhm, так как выводы ~INT0 и ~INT1 у V9990 с открытым стоком (коллектором?). AVR-контроллер оказался шустрым (16 MHz), но софтварно ~WAIT городить не хотелось, поэтому выставлены несколько Wait-State'ов для External SRAM (тобишь – регистров V9990).

Результаты

Привожу здесь несколько HI-Quality фотографий, которые на мой взгляд интересны:







Заключение

Хотелось бы подвести итоги на *данный* момент времени. Я – один из тех, кто удачно запустил и начал осваивать видеочип V9990. Несмотря на возникшие трудности, они были успешно преодолены не только мной, но и теми людьми, которые мне помогали (и помогут в будущем - надеюсь)!

Настоящее положение дел такое: плата находится в плачевном состоянии, но переразводить и делать релиз ещё пока рано! Этому есть несколько оснований, а именно:

1) Не выяснена *явно* причина МЕГА-глюка (загрязнение содержимого видеопамяти в момент интенсивной работы блиттера)

2) Не устаканена *окончательная* принципиальная схема VideoMonstr'a. Мне хочется ещё поэкспериментировать… Применить ICS501, найти кварцевый осциллятор на 4.433618 MHz, поставить CXA1645 или CXA2075 и много чего ещё, чего не додумал или не вспомнил…

3) Не разведена *корректно* печатная плата устройства, не стандартизирована шина (при использовании VideoMonstr'a как модуля).

4) Красиво изготовить печатную плату (двуслойка с переходными отверстиями и т.п.) вручную *невозможно* – нужно заказывать на заводе.

Поэтому хочу сказать – проект и наработки по нему интересны, но он пока ещё зелёный. Поэтому точка над V9990 ещё не поставлена! **J**

Apparatchik Romanich
25.04.07