

Таблица

Контакт	Цепь	Контакт	Цепь
A1	+5V	B1	GND
A2	F	B2	A11
A3	A12	B3	A10
A4	A13	B4	A9
A5	A14	B5	A8
A6	D7	B6	A7
A7	D6	B7	A6
A8	D5	B8	A5
A9	D4	B9	A4
A10	D3	B10	A3
A11	D2	B11	A2
A12	D1	B12	A1
A13	D0	B13	A0
A14	/CSS	B14	/WRS
A15	SOUND1	B15	/INT
A16	SOUND0	B16	GND
A17	/WRV	B17	/CSV
A18	/VCS'	B18	VA10'
A19	/VA13	B19	VA6
A20	VA7	B20	VA5
A21	VA8	B21	VA4
A22	VA9	B22	VA3
A23	VA10	B23	VA2
A24	VA11	B24	VA1
A25	VA12	B25	VA0
A26	/OEV	B26	VD0
A27	VD7	B27	VD1
A28	VD6	B28	VD2
A29	VD5	B29	VD3
A30	VD4	B30	+5V

Примечание. Знак "/" означает сигнал с низким активным уровнем

Особенностью ADR-LOG-картриджа (рис.4) является дешифратор DS, выполненный на ... счетчике DD1 GD74LS161A (аналог K555IE10). Нестандартность ситуации в том, что счетчик используется в качестве регистра хранения. Его входы подключаются к младшим разрядам шины адреса A0-A3, а выходы – к старшим разрядам S-ROM и V-ROM.

Переключение банков происходит программно по команде записи любого байта в S-ROM, при этом ЦП выставляет требуемый адрес и активизирует сигналы /WRS и /CSS. Разумеется, запись в ПЗУ физически не происходит, однако по сигналу /WRS срабатывает дешифратор банков.

Попытка определить количество банков визуальным путем может оказаться неудачной. Например, на рис.4 видно, что N=8, M=8, поскольку вывод 11 DD1 соединяется только с DD2. Однако в другом картридже с похожей схемой было обнаружено 8 банков S-ROM 16 кбайт, 4 «виртуальных» банка S-ROM 32 кбайт и 8 банков V-ROM. Разгадка заключается во внутренней структуре микросхемы-«капельки» S-ROM, внутри которой, очевидно, содержатся дополнительные логические схемы дешифрации.

Емкость бескорпусных микросхем ПЗУ часто можно вычислить по надписям на печатной плате. Например, маркировка микросхемы DD2 (рис.4): «GS» – фирма «Gold-Star»; «1M» – общий объем памяти 1 Мбит; «16» – емкость одного банка памяти. Заметим, что размер памяти для ПЗУ принято выражать в битах. Чтобы получить размер памяти в байтах, необходимо разделить искомую величину на 8.

(Продолжение следует)

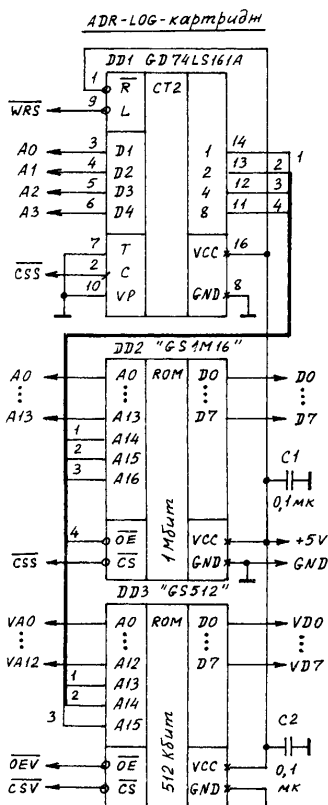


рис. 4

мер, в приставке «Dendy Junior-II» встречается 80-выводная микросхема-«капелька» UM6561A, расположенная этажерочным модулем на отдельной печатной плате размерами 40x40 мм.

На рис.1 показана карта распределения адресного пространства «Dendy», составленная на основании раритетных публикаций [2-5].

Адресное пространство ЦП составляет 64 кбайт. Адреса 0-7Fh занимает внутреннее основное ОЗУ, в котором размещают стек и переменные. В ранних моделях «Dendy» это ОЗУ было выполнено на отдельной корпусной микросхеме типа HM6116LK-70 (Hitachi) объемом 2 кбайт. Вследствие неполной дешифрации адресов, доступ к основному ОЗУ с одинаковым успехом можно производить еще через три окна: 800-0FFFh, 1000-17FFFh, 1800-1FFFh.

Адреса 2000-3FFFh занимает область портов ввода/вывода ВП, причем наиболее активно используются первые 8 ячеек (2000-2007h). Далее располагается область портов ввода/вывода периферийных устройств (4000-5FFFh), а именно, музыкального процессора, двух джойстиков и светового пистолета. Завершает первую половину адресного пространства резервное место (6000-7FFFh).

Верхняя область адресов полностью отдана под ПЗУ картриджа (S-ROM) емкостью 32 кбайт. Для расширения памяти применяется стандартный прием – «вклеивание» дополнительных страниц или, по-другому, банков памяти через специальные дешифраторы. Типичное число банков S-ROM – от 3 до 8 при размере банка 16 или 32 кбайт. Встречаются картриджи с числом банков до 50, однако это «виртуальные» (а не фи-

зические) построения, поскольку они образуются математическим сочетанием небольшого числа банков емкостью 16 кбайт.

ВП имеет собственное адресное пространство (не путать с гарвардской архитектурой микропроцессоров!), в самом начале которого расположено видеоПЗУ картриджа (V-ROM, 0-1FFFh). Расширить емкость можно по аналогии с ЦП через «вклеивание» дополнительных банков памяти емкостью 4 или 8 кбайт. Иногда вместо видеоПЗУ в картридже можно обнаружить КМОП-видеоОЗУ (V-RAM) объемом 2, 8 или 32 кбайт.

Адреса памяти 2000-2FFFh отводятся под 4 экранные области, которые обслуживает неполовностью адресуемое внутреннее видеоОЗУ емкостью 2 кбайт. В ранних моделях «Dendy» видеоОЗУ было выполнено на отдельной корпусной микросхеме, аналогично основному ОЗУ. По адресам 3000-3FFFh располагается область регистров цвета, доступная только на запись.

Итак, в Dendy-картридже может находиться: основное ПЗУ S-ROM (8-512 кбайт), видеоПЗУ V-ROM или видеоОЗУ V-RAM (2-64 кбайт), а также дешифратор банков (DS). Не все из перечисленных составляющих присутствуют в каждом картридже, поэтому введем классификацию.

Классификация картриджей

Dendy-картриджи можно условно разделить на две группы:

1) простые SV-картриджи (один банк S-ROM, один банк V-ROM или V-RAM, без DS); 2) страничные картриджи (N банков S-ROM, M банков V-ROM или V-RAM, дешифратор DS).

В свою очередь, страничные картриджи подразделяют на RES-картриджи (переключение банков по сбросу), ADR-картриджи (переключение банков по шине адреса), DAT-картриджи (переключение банков по шине данных). Кроме того, страничные картриджи независимо от типа могут иметь дешифратор адресов, выполненный на дискретных логических элементах (LOG-) или на программируемых логических матрицах (PLM-).

Картридж соединяют с приставкой через 60-контактный торцевой печатный разъем. Привязка сигналов к лапелям разъема приведена в таблице. Контакты B2-B14, A2-A14 относятся к S-ROM, а B17-B29, A17-A30 – к V-ROM. Перечислим назначение сигналов, наиболее часто используемых в картриджах:

A0-A14, VA0-VA12 – шины адреса ЦП и ВП; D0-D7, VD0-VD7 – шины данных ЦП и ВП; /WRS, /WRV – чтение/запись S-ROM и V-ROM («1/0»);

/CSS – запрет/выбор S-ROM («1/0»); /CSV, /OEV – запрет/выбор V-ROM («1/0»);

F – синхросигнал ЦП.

Поскольку сигналы, относящиеся к S-ROM, физически сгруппированы возле младших, а V-ROM – возле старших порядковых номеров контактов разъема, то становится понятным, почему микросхема-«капелька» S-ROM обычно располагается в левой, а V-ROM – в правой части картриджа (рис.2).

Электрические схемы картриджей

Простой SV-картридж (рис.3) можно рассматривать, как два независимых ПЗУ DD1 и DD2 емкостью соответственно 32 и 8 кбайт. В современных картриджах этот вариант встречается редко.

ПОДКЛЮЧЕНИЕ DENDY-КАРТРИДЖЕЙ К IBM PC

(Продолжение. Начало см. в РА 4/2000)

С.М. Рюмик, г. Чернигов

Следующая разновидность – **RES-LOG-картридж (рис.5)**, в котором выбор одной из двух игр происходит циклически при каждом нажатии кнопки «RESET» приставки. На разъем подключения картриджа, к сожалению, не выведен сигнал сброса, поэтому приходится искать обходные пути.

Переключением банков заведует триггер DD2. На его синхровход во время игры поступают импульсы F через детектирующую цепочку VD1, R1, C1. При нажатии/отпускании кнопки сброса подача импульсов F кратковременно прекращается, вызывая перепад напряжения на входе DD2 и переключение триггера в противоположное состояние. Сигнал с выхода 5 микросхемы DD2 управляет выбором одного из двух банков S-ROM, объемом 32 кбайт каждый.

ВидеоПЗУ в данной схеме отсутствует. Его место занимает видеоОЗУ (V-RAM), выполненное на бескорпусной микросхеме DD3 емкостью 8 кбайт. На печатной плате обычно делают универсальную разводку дорожек, оставляя место для 28-выводной корпусной микросхемы типа SRM2064M-15 (Seiko Epson) – аналог KP537PY17.

Картриджи с PLM – это наиболее сложные для анализа разновидности. В общем случае они могут иметь префикс ADR-PLM-, DAT-PLM-, RES-PLM-. При наличии DAT-PLM-картриджа, состоящего из одной БИС (рис.6), входы внутреннего дешифратора банков подключают к одному или нескольким разрядам шины данных D0-D7, а выходы – к старшим адресным разрядам S-ROM и V-ROM. Активизация дешифратора осуществляется сигналами /WRS, /CSS, при этом формально происходит еще и запись в ПЗУ.

В отличие от ранее рассмотренного ADR-LOG-картриджа, в данном случае может наблюдаться конфликт на шине данных. Действительно, ЦП, переключая по шине данных дешифратор, одновременно пытается записать этот же код в ПЗУ, которое при наличии низкого уровня сигнала /CSS переходит в режим чтения. Налицо конфликтная ситуация, которую может разрешить только программист выбором адреса записи, уже содержащего требуемый код.

Следует отдать должное разработчикам «Dendy», применившим это оригинальное техническое решение. Получается, что для переключения большого количества банков памяти физически используется всего одна (!) дополнительная линия сигнала /WRS. Некоторое неудобство при программировании с лихвой окупается максимальным воздействием смены банков.

Рассмотренные примеры не исчерпывают все разновидности картриджей. О внутреннем устройстве микросхем-«капелек» можно говорить лишь в предположительном тоне. Например, внутри могут находиться дополнительные логические элементы, в частности, модифицирующие номер банка в зависимости от игровых условий и т.д.

Методы считывания информации с Dendy-картриджей

Поскольку «осевой балкой» Dendy-кар-

триджа являются два независимых ПЗУ, то задача сводится к получению их карт прошивки. Для простых SV-картриджей решение банально – следует изготовить два кабеля-переходника и подключить через них по очереди S-ROM и V-ROM к любому программатору. К сожалению, таких картриджей немного, они, как правило, содержат одну незатейливую игру.

Для картриджей «покруче» такой подход не годится, поскольку программатор не в состоянии самостоятельно переключать банки памяти. Первое упоминание об успешном считывании информации из ПЗУ Dendy-картриджей на бытовой магнитофон датируется началом 1995 г. [2]. Позже был разработан специальный адаптер из 10 микросхем, подключаемый к системной шине компьютера «ZX-SPECTRUM» и позволяющий записывать содержимое картриджа на дискету [3]. Следующий шаг – применение в качестве носителя информации видеомангитфонной ленты [6]. Широкому внедрению адаптеров помешали три фактора: закаты SPECTRUM-совместимых компьютеров, закрытость программного обеспечения, включая прошивки ПЛИС, и малый тираж литературы.

Попробуем создать «ремикс» адаптера, но для IBM-совместимых компьютеров. Заботимся о том, чтобы у разработки не было мешающих внедрению факторов. А именно, программное обеспечение должно быть открытым, схема несложной, а в качестве входного необходимо применить стандартный порт, позволяющий использовать как архаичный IBM-286, так и мощный Pentium-II. Попутно заметим, что вставлять напрямую

Dendy-картридж в слот шины ISA материнской платы IBM PC нельзя, хотя физически можно. Это приведет в лучшем случае к выходу из строя картриджа, а в худшем – компьютера.

Электрическая схема считывающего устройства (СУ) показана на **рис.7**. Основой является «неуязвимая» БИС DD1 KP580BB55A, которая через разъем XP1 получает информацию от параллельного порта принтера (LPT-порт) и формирует сигналы выборки адресов A0-A14 (выводы PA0-PA7, PB0-PB6) и управления (выводы PC0-PC7). Dendy-картридж подключают к розетке XS1. Выходные сигналы картриджа D0-D7 и VD0-VD7 через мультиплексоры DD2, DD3 поступают на разъем XP1 в IBM PC.

Светодиод HL1, кроме своей индикаторной функции, обеспечивает уровень лог. «1» для входа /RD DD1. Цепочка C2, R2 обеспечивает начальный сброс СУ, уменьшая вероятность конфликта между сигналами DATA0-DATA7 компьютера и D0-D7 микросхемы DD1. Такая ситуация, хотя и теоретически, но возможна в первый промежуток времени после включения питания.

Применение БИС DD1 в данной схеме не случайно. Во-первых, это позволяет существенно сократить число радиоэлементов, во-вторых, осуществить надежную фиксацию информации в буферных регистрах портов PA, PB, PC.

Нюанс в том, что пользователь при работе с LPT-портом не является его единоличным собственником. Нормальная работа при обращении к LPT-порту периодически нарушается из-за системных прерываний, что можно заметить логическим пробником, регистрирующим кратковременные импульсы в самых неожиданных комбинациях на разъеме XP1. Эти импульсы не позволяют напрямую управлять процессами и требуют обязательного применения во входных цепях регистров хранения, причем с управлением по фронту сигнала, а не по уровню. Именно для такого случая, как нельзя кстати, подходит микросхема KP580BB55A, имеющая вдобавок еще и внутренние регистры хранения информации выходных портов.

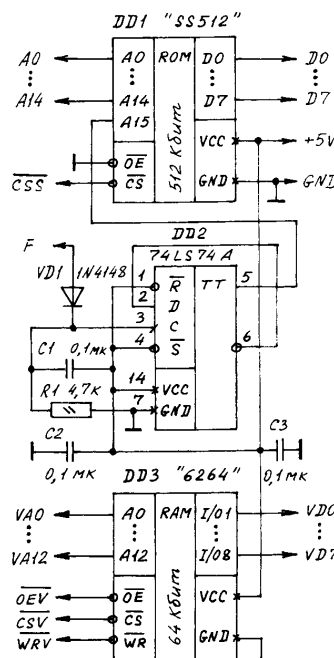


рис. 5

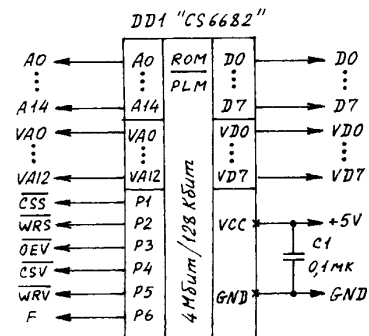


рис. 6

"CARTRIDGE DENDY"

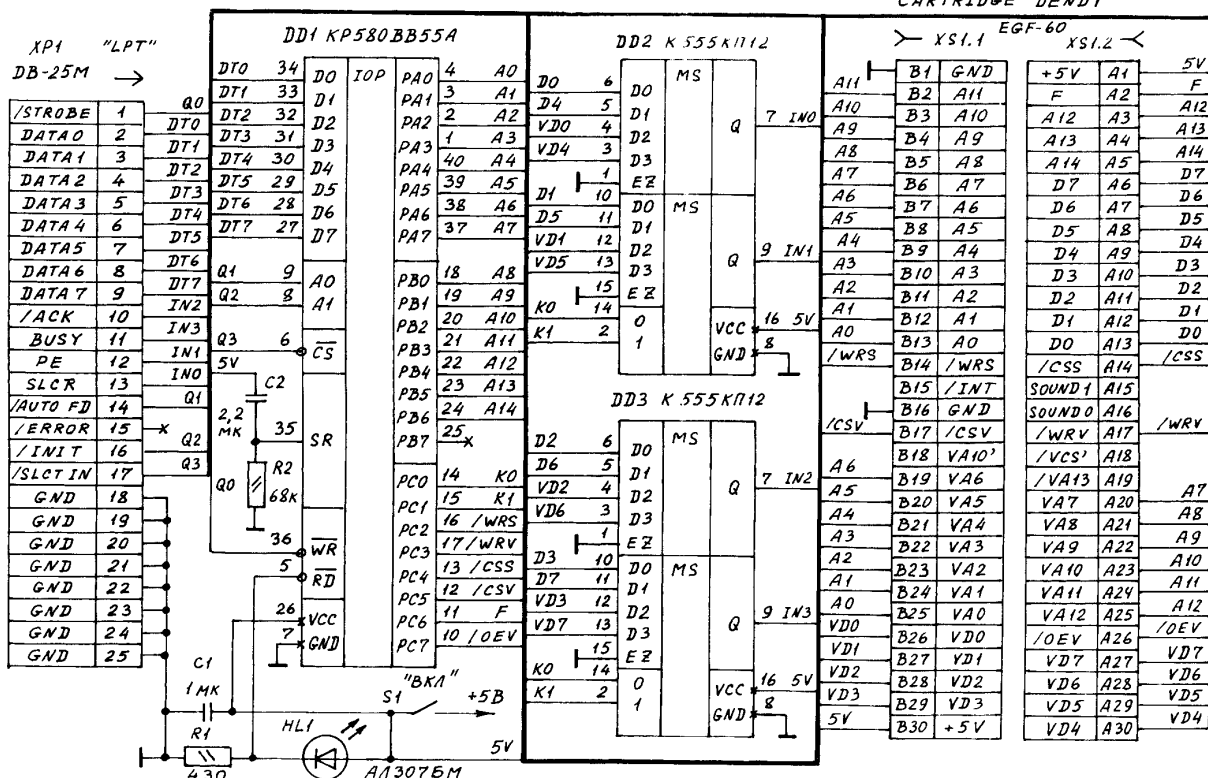


рис. 7

Конструкция и детали

СУ выполняют в виде отдельной конструкции, соединяемой с разъемом LPT-порта ленточным кабелем длиной не более 1,5 м. Внешний вид конструкции зависит от розетки XS1, в качестве которой удобно использовать слот от неисправной «Dendy». Во избежание повреждений, его следует не выпаивать, а аккуратно вырезать вместе с частью печатной платы. Применять слоты от 8-разрядной шины ISA здесь не рекомендуется, поскольку они имеют 62, а не 60 ламелей.

Монтаж проще вести тонким проводом, установив микросхемы DD1-DD3 на панельки. Резисторы типа ОМЛТ-0,125, конденсаторы — КМ-66. Переключатель S1 — любой тумблер, удобный по конструкции. Светодиод — любого цвета, обеспечивающий падение напряжения не более 2 В. Вилка XP1 импортная, так называемая «принтерная», или СМП101-25В.

Как показывает практика, схема не критична к длине соединительных проводов, качеству монтажа и экранировке. Питание от внешнего источника напряжением 4,9–5,1 В, рассчитанного на ток не менее 200 мА. В случае неустойчивой работы схемы следует провести стандартные мероприятия: установить в точке подключения проводов питания электролитический конденсатор К50-35 емкостью 50–100 мкФ; применить экранировку соединительного кабеля; между каждым из выводов 1-9, 14, 16, 17 разъема XP1 и цепью +5 В установить резисторы ОМЛТ-0,125 сопротивлением 1–4,7 кОм.

Карtridge следует вставлять (вынимать) в (из) розетку XS1 только при выключенном тумблере S1. Не рекомендуется отсоединять кабель от принтерного порта при работающем компьютере.

Программная часть

Схема СУ без наличия программного обеспечения является бесполезным «конгломератом» кремния, пластмассы и железа. Нельзя ограничиться фразой наподобие: «Подготовленные радиолюбители могут написать программу самостоятельно». Правила хорошего тона требуют от разработчика привести алгоритм работы, побитовую раскладку портов, краткую демоверсию программы, написанную на общедоступном языке программирования. Демоверсия — это не парадный инструмент, а «рабочая лошадка», позволяющая с минимальным сервисом выполнять основную функцию устройства.

Итак, с точки зрения программиста, LPT-порт представляет собой блок из трех расположенных друг за другом 8-разрядных регистров: регистра данных (РД, запись/внутреннее чтение), регистра статуса (РС, чтение) и регистра управления (РУ, запись/внутреннее чтение). Напомним, что РД считается базовым. Термин «внутреннее чтение» означает наличие простого однонаправленного LPT-порта. Рассчитывать на то, что в IBM-286 можно без проблем встроить улучшенный двунаправленный параллельный порт формата EPP или ECP, наивно.

На 25-контактный принтерный разъем IBM PC выведены сигналы 8 разрядов РД, 5 разрядов РС и 4 разряда РУ (табл.2). В схеме СУ задействованы все сигналы, кроме РС:3, причем РД и РУ работают только на запись, а РС — на чтение.

Как известно, параллельным портам в IBM PC присвоены аббревиатуры LPT1, LPT2, LPT3. Между собой они различаются местоположением базового адреса РД. Стандартные адреса параллельных портов — это 378, 379, 37Ah (LPT1) и 278, 279, 27Ah (LPT2). Если в системе присутствует «доисто-

Таблица 2

Контакт LPT-разъема	Сигнал	Разряд регистра	Примечание
1	/STROBE	РУ:0	-
2	DATA0	РД:0	+
3	DATA1	РД:1	+
4	DATA2	РД:2	+
5	DATA3	РД:3	+
6	DATA4	РД:4	+
7	DATA5	РД:5	+
8	DATA6	РД:6	+
9	DATA7	РД:7	+
10	/ACK	РС:6	+
11	BUSY	РС:7	-
12	PE	РС:5	+
13	SLCR	РС:4	+
14	/AUTO FD	РУ:1	-
15	/ERROR	РС:3	-
16	/INIT	РУ:2	+
17	/SLCT IN	РУ:3	-
18-25	GND		

* Знаком "-" отмечены линии регистров, которые аппаратно инвертируются внутри IBM PC.

рический» дисплейный адаптер MDA/HERCULES, имеющий на плате собственный принтерный порт с адресами 3BС, 3BD, 3BEh, этому порту BIOS присваивает обозначение LPT1, а вышеперечисленным стандартным — LPT2 и LPT3 соответственно. Почему так? Со времен выпуска первой DOS операционная система IBM PC при инициализации запрашивает информацию о наличии параллельных портов в строго определенном порядке, а именно, 3BС, 378, 278h.

(Продолжение следует)

ПОДКЛЮЧЕНИЕ DENDY-КАРТРИДЖЕЙ К IBM PC

(Окончание. Начало см. в РА 4,5/2000)

С.М. Рюмик, г. Чернигов

Программа-драйвер

На листинге 1 представлена программа-драйвер для СУ, написанная на языке TURBO C (версия 2.0) и предназначенная для считывания информации Dendy-картриджей в память IBM PC. Язык C (C++) мог бы стать основой для периферийных устройств подобного класса ввиду гибкости, компактности исходного текста и хорошего быстродействия компиляторов. Следует лишь избегать применения нестандартных библиотек функций и не экономить на комментариях.

На рис.8 и 9 приведены побитовые раскладки задействованных регистров соответственно для LPT-порта и для БИС KP580BB55A.

Программа содержит основную часть «main()» и три функции: «kc()», «ff()», «out(p,y)». Для прочтения очередной ячейки S-ROM или V-ROM необходимо:

занести через РД и РУ LPT-порта младшую часть адреса A0-A7 в регистры PA0-PA7 DD1;

занести аналогично старшую часть адреса A8-A12 (для V-ROM) или A8-A14 (для S-ROM) в регистры PB0-PB6 DD1;

установить режим чтения младшего nibбла (4 разряда) шины данных картриджа с занесением соответствующей информации в разряды PC0-PC7 DD1;

прочитать через PC LPT-порта 4 младших разряда шины данных картриджа;

аналогично установить режим чтения старшего nibбла и прочитать 4 старших разряда шины данных картриджа;

«склеить» программно старшие и младшие разряды для получения байта считанной информации.

Приведенный алгоритм работы реализован в программе функцией «kc()», в обязанности которой входит также подсчет контрольной суммы блока и вывод информации в виде файлов. Функция «ff()» программно формирует одиночный импульс отрицательной полярности сигнала /WRS. Функция «out(p,y)» позволяет сократить текст листинга.

Программа самостоятельно определяет

размер банка S-ROM от 8 до 32 и V-ROM от 2 до 8 кбайт, а также распознает ситуацию отсутствия картриджа и наличия V-RAM вместо V-ROM. Время поиска одного банка и подсчета контрольной суммы невелико. Оно мало зависит от быстродействия компьютера. Например, для IBM-286 это примерно 2,1 с; для AMD-K5-166 – 1,4 с; для Pentium-200-MMX – 1,1 с. При желании быстродействие можно повысить в десятки раз, используя алгоритм выборочного подсчета контрольной суммы, однако при этом повышается опасность пропуска очередного банка.

Представленная версия программы для упрощения не использует автоматический поиск банков. Переключать банки можно в ручном режиме тремя способами:

1) через имитацию сброса приставки формированием уровней «0-1-0» на входе F картриджа (разряд PC6 DD1).

2) через произвольный адрес, указываемый пользователем. При этом производится выбор ячейки и установка уровней «1-0-1» сигнала /WRS.

3) через произвольный код на шине данных. Первоначально ищется ячейка, содержащая требуемый код, затем выбирается адрес найденной ячейки и устанавливаются уровни «1-0-1» сигнала /WRS.

Поиск банков – процесс творческий. Нет смысла перебирать все по очереди адреса ячеек. Для практических целей достаточно ввести 15 адресов, пользуясь формулой: $A=2K$, где A – вводимый адрес; $K=0, \dots, 14$. После ввода очередного адреса анализируют, изменилась ли контрольная сумма банка и его размер. Если «да», то разряд с номером K участвует в дешифрации банка. Обычно задействовано не более 4 разрядов, поэтому на следующем этапе вводят адреса с их всевозможными комбинациями.

Аналогичную процедуру применяют при поиске переключений по шине данных. Последовательно вводятся 8 кодов, вычисляемых по формуле: $D=2P$, где D – вводимый код; $P=0, \dots, 7$.

Результат работы программы хранится в двоичных (не текстовых) файлах, записыва-

емых в текущий каталог диска. Имена файлов отдельно для S-ROM и V-ROM задает пользователь. Число символов в имени должно быть не более 12, включая точку и расширение. Допускается точка и расширение не указывать. Идентификация различных банков производится по контрольной сумме. В данном случае это 6- или 7-разрядное десятичное число, полученное простым суммированием всех байтов файла.

Объем скомпилированной «.exe» программы листинга 1 составляет около 55 кбайт, что позволяет запускать ее не только с жесткого, но и с гибкого диска.

Анализ игровых программ для «Dendy»

Что делать после получения кодов прошивок S-ROM и V-ROM? Задача-минимум – рассортировать банки по играм. Подсказка: в «многоигровках» каждая игра использует один банк S-ROM и один V-ROM, не переключаясь во время работы. В принципе можно методом последовательного перебора вариантов подобрать подходящую пару S-ROM и V-ROM. Бывает, что легче найти аналогичную игру на другом, более мягком для анализа, картридже.

Ускорить работу поможет просмотр текстов, содержащихся в прошивках ПЗУ через любой редактор, например, Norton Commander. По обрывкам фраз в S-ROM и V-ROM несложно отличить одну игру от другой, увидеть пункты меню, результаты в таб-

Назначение битов регистров LPT-порта

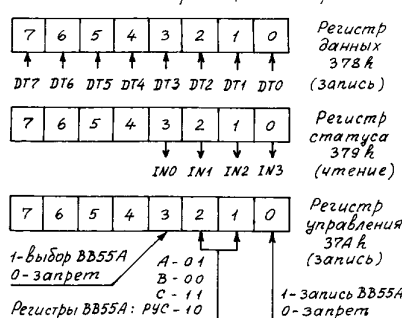


рис. 8

Назначение битов регистров KP580BB55A

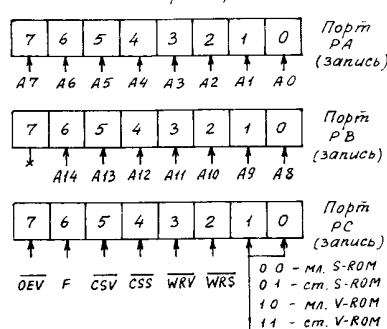


рис. 9

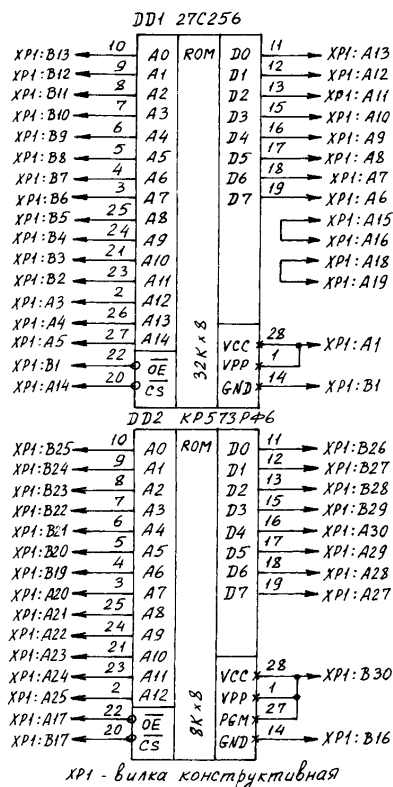


рис. 10

ЛИСТИНГ 1

```

/* СЧИТЫВАНИЕ DENDY-КАРТРИДЖА В IBM PC (Рюмик С.М.) */
#include <stdio.h> /* Журнал "Радиолюбитель", 2000 г.
#include <dos.h> /* BORKLAND Turbo C, версия 2.0
#define MAX 32768 /* Максимальный размер банка
char srom[MAX]; /* Массив данных банка S-ROM
unsigned a; /* Базовый адрес LPT-порта

main()
{
extern char srom[MAX]; /* Внешний массив S-ROM
extern unsigned a; /* Внешний базовый LPT-адрес
int ch; /* Символ, вводимый с клавиатуры
unsigned z, dc, dd;

printf ("Укажите базовый адрес LPT-порта: ");
while ((ch = getch()) < 20 || ch > 255) /* Опрос
if (ch == 0) a = 0x278; /* Клавиша <=
else a = (ch == 50) ? 0x278 : 0x2BC; /* Клавиша <2>, <3>
printf ("Выбран LPT-порт с адресом: ");
printf ("%Xh", a); /* Вывод адреса
outportb (a+2,12); out (0x80,13); /* Режим-0 B555A
kc(); /* Чтение картриджа, подсчет KC

puts ("Переключение банков по сбросу (Y/N)?");
while ((ch = getch()) != 89 || ch != 121) /* Опрос
{
out (0x0C,15); /* Сигнал F=1
out (0x0C,15); /* Сигнал F=1
out (0x0C,15); /* Сигнал F=0
kc(); /* Чтение картриджа, подсчет KC
puts ("Произвести очередной сброс (Y/N)?");
}

puts ("Переключение банков по шине адреса (Y/N)?");
while ((ch = getch()) != 89 || ch != 121) /* Опрос
{
printf ("Введите десятичный код: ");
scanf ("%u", &dc); /* Ввод адреса
if (dc >= MAX) puts ("Очень большое значение!");
else { printf ("Введен адрес: %u\n",dc);
ff((dc)); /* Переключение банков
kc(); /* Чтение картриджа, подсчет KC
}
puts ("Произвести ввод другого адреса (Y/N)?");
}

puts ("Переключение банков по шине данных (Y/N)?");
while ((ch = getch()) != 89 || ch != 121) /* Опрос
{
printf ("Введите десятичный код: ");
scanf ("%u", &dd); /* Ввод кода
if (dd > 255) puts ("Очень большое значение!");
else { for (z=0; srom[z] != dd) z++;
if (z == MAX)
{ printf ("Код %u не найден!\n",dd);
else { printf ("Введен код %u\n",dd);
printf ("По адресу: %u\n",z);
ff(z); /* Переключение банков
kc(); /* Чтение, подсчет KC
}
}
printf ("Произвести ввод другого значения: ");
puts ("Данных (Y/N)?");
}
puts ("\t Работа программы завершена."); /* Стоп */

/* ФУНКЦИЯ: ЧТЕНИЕ СОДЕРЖИМОГО КАРТРИДЖА, ПОДСЧЕТ
КОНТРОЛЬНЫХ СУММ (KC) ЗАПИСИ БАНКОВ S-ROM, V-ROM
kc()
{ FILE *sout, *vout; /* Указатели на файлы записи
extern char srom[MAX]; /* Внешний массив S-ROM
extern unsigned a; /* Внешний базовый LPT-адрес
int ch; /* Символ, вводимый с клавиатуры
char dn, ds, ds; /* Массив данных банка V-ROM
char vname[12], vname[12]; /* Имя (1-12) символов
unsigned j, i; /* Вспомогательные переменные
unsigned long skc[10] = {0,0}; /* Контроль. сумма S-ROM
unsigned long vkc[10] = {0,0,0,0}; /* Кон. сумма V-ROM

/* Чтение и подсчет контрольной суммы S-ROM */
for (i=0; i<2; i++) /* Два банка S-ROM по 16Кб
for (j=1; j<=MAX/2; j = (j+1)*MAX/2; j++)
{ dn=j/256; ds=j-dn*256; /* Перевод в hex-код
out (ds,11); out (dn,9); /* Ввод адреса
out (0x0C,15); /* Сигнал F=1
dat = (0x80 inportb(a+1)) >> 4; /* Чтение ниббл
out (0x0C,15); /* Перевод на старший ниббл
srom[j] = (0x80 inportb(a+1)) & 0xF0; dat;
skc[i] += srom[j]; /* Подсчет контрол. суммы
if (srom[j] == 0xFF) srom[i]++; /* Нет картриджа
}
if (skc[0] != skc[1]) srom[i]++; /* Банк S-ROM 2Кб
else { srom[i] = MAX/8; skc[i]++; /* Банк S-ROM 16Кб

/* Чтение и подсчет контрольной суммы V-ROM */
for (i=0; i<4; i++) /* Четыре банка V-ROM по 2 Кб
for (j=1; j<=MAX/16; j = (j+1)*MAX/16; j++)
{ dn=j/256; ds=j-dn*256; /* Перевод в hex-код
out (ds,11); out (dn,9); /* Ввод адреса
out (0x16,15); /* Младший ниббл V-ROM
dat = (0x80 inportb(a+1)) >> 4; /* Чтение ниббл
out (0x17,15); /* Перевод на старший ниббл
vrom[j] = (0x80 inportb(a+1)) & 0xF0; dat;
vkc[i] += vrom[j]; /* Подсчет контрол. суммы
if (vrom[j] == 0xFF) vrom[i]++; /* Наличие V-ROM
}
if ((vkc[0]+vkc[1]) != (vkc[2]+vkc[3])) vkc[i]++; /* Банк V-ROM 8Кб
else if (vkc[0] != vkc[1]) /* Банк V-ROM 4Кб
{ skc[2]=skc[3]=0; vkc[i]++; /* Банк V-ROM 4Кб
else { skc[1]=skc[2]=skc[3]=0; vkc[i]++; /* Банк V-ROM 2Кб

/* Вывод результатов, запись файлов S-ROM и V-ROM */
if (srom[i] == MAX) puts ("Картридж отсутствует!");
else
{ printf ("Найдено: S-ROM %dKb (k=4lu);", i);
if (vrom[i] <= (MAX/4-10)) printf ("V-ROM %dKb\n", i);
else printf ("V-ROM %dKb (k=4lu)\n", i);
vax/204; vkc[0]+vkc[1]+vkc[2]+vkc[3];
puts ("Записать (Y/N)?");
if ((ch=getch()) != 89 || ch != 121) /* Опрос */
{ printf ("Введите имя файла для записи S-ROM: ");
scanf ("%s", sname); /* Ввод имени файла S-ROM
if (sout = fopen (sname, "wb")) == NULL)
{ puts ("Невозможно создать файл!");
exit(1); /* Аварийный выход
}
for (j=0; j < srom; j++) puts (srom[j],sout);
fclose (sout); /* Закрытие файла S-ROM

if (vrom[i] <= (MAX/4-10))
{ printf ("Введите имя файла для записи V-ROM: ");
scanf ("%s", vname); /* Ввод имени файла V-ROM
if (vout = fopen(vname, "wb")) == NULL)
{ puts ("Невозможно создать файл!");
exit(1); /* Аварийный выход
}
for (j=0; j < vrom; j++) puts (vrom[j],vout);
fclose (vout); /* Закрытие файла V-ROM
}
}
outportb (a+2,12); out (0x80,13); /* Режим-0 B555A

/* ФУНКЦИЯ ПЕРЕКЛЮЧЕНИЯ ДЕШИФРАТОРА БАНКОВ */
ff(unsigned p, unsigned a) /* Адрес ячейки j в hex-коде
char dn, ds; /* Внешний базовый LPT-адрес
extern unsigned a; /* Внешний базовый LPT-адрес

dn=j/256; ds=j-dn*256; /* Перевод в hex-код
out(ds,11); out(dn,9); /* Установка адреса
out(0x0C,15); /* Сигнал F=1, -CSS=0
out(0x0C,15); /* Сигнал F=0, -CSS=0
out(0x0C,15); /* Сигнал F=0, -CSS=1
out(0x0C,15); /* Сигнал F=1, -CSS=1
}

/* ФУНКЦИЯ ЗАПИСИ КОДОВ В РУС, А-, В-, С-порты B555A */
out(unsigned p, unsigned a) /* Входящие параметры:
p - вводимый код; y - выбор регистров B555A
extern unsigned a; /* Внешний базовый LPT-адрес

outportb (a,p); /* Загрузка кода в базовый адрес
outportb (a+2,y); /* Запись кода в регистр B555A
outportb (a+2,y-1); /* Хранение кода в регистре

```

лице рекордов, имена создателей игры и даже хакерские «графити», выполненные в виде причудливых китайских иероглифов.

Далее необходим логический анализ. Производить расшифровку программы вручную малопривлекательно. К счастью, для IBM PC написаны кроссассемблеры, мониторы и отладчики, ориентированные на среду знаменитого микропроцессора MC6502 (Motorola). «Знаменитого», поскольку он широко применялся в первых американских компьютерах личного пользования: «APPLE-II» (1977 г.) фирмы «Apple Computer Corp.», «PET» (1977 г.), «VIC-20» (1980 г.) фирмы «Commodore Business Machines»; «Atari 400/800» (1978 г.) фирмы «Atari» [7]. Система команд процессоров 6527, 6561 в целом совпадает с 6502 [2]. Следовательно, нет ограничений для работы с широкодоступными программами «PseudoSam 65 assembler» (a65.com), «SVASMO2 cross-assembler» (svasmo2.com), «Cross-reference 6502» (6502asm.com) [8].

Каждый картридж требует для анализа индивидуального подхода. Дать общей рекомендации не представляется возможным, многое зависит от хакерского (в хорошем смысле слова) таланта, отпущенного природой пользователю.

Первая задача – определить ведущий банк, в котором расположено начало программы или меню. Поскольку дешифратор банков при начальном включении питания может самопроизвольно устанавливаться в любое состояние, то в каждом банке программист должен предусмотреть процедуру приоритетного перехода в основной (ведущий) банк.

Для начала анализируют информацию, содержащуюся в точке старта. Абсолютные адреса старта «холодного» (при включении питания) и «теплого» (при нажатии кнопки сброса видеоприставки) совпадают. Это ячейка 0FFFCh.

На истинное начало игры, размещенной в конкретном банке, часто указывает вектор маскируемого прерывания (адрес 0FFFh). Многие игры начинаются процедурой инициализации, которая выглядит в 16-ричном (hex) коде примерно так: 78 A9 00 8D 00 20 8D 00 21. Эту комбинацию легко найти внутри файла встроенным редактором Norton Commander.

Если в начале программы обнаружился длинный цепочка холостых операций (код 00h), значит, хакер «ампутировал» загрузочный блок, а игра инициализируется где-то в другом месте, как правило, в банке, содержащем программу работы с начальным меню.

В «многоигровках» при наличии меню с несколькими сотнями вариантов игр, процедура включения ведущего банка часто переносится в область основного ОЗУ по адресам 100-700h. Далее из ОЗУ включается ведущий банк и передается управление на истинный адрес старта.

Разумеется, несмотря на многообещающие надписи на обложке картриджа, «999» игр в чистом виде в картридж не помещается. Секрет в том, что для каждого варианта игры в тело программы предварительно заносятся коды, изменяющие нормальный ход действия. Это может быть переход на

другой уровень, наделение главного героя бессмертием, а иногда и плоская шутка в виде загорченного экрана. Программы введения «волшебных» кодов (POKE) очень короткие, их действительно может быть несколько сотен, что создает иллюзию большого количества игр.

Дополнительные сведения по нахождению точек переключения банков и описание системы команд ЦП приведено в [2-5, 9].

Коррекция программ

Остается ответить на последний вопрос – как устранить дефекты игры? Предположим, в результате кропотливой работы выделены банки S-ROM и V-ROM, отвечающие за конкретную игру, а также найдено место, которое, вероятнее всего, нарушает нормальный ход течения программы. Теперь можно поупражняться в ассемблерном программировании и составить «patch» (в переводе с англ. – заплатка) – кодовый блок, восстанавливающий работоспособность.

Любую теорию следует проверять на практике. Следовательно, отредактированные прошивки S-ROM и V-ROM необходимо через программатор записать в микросхемы ПЗУ. Для S-ROM подойдет ПЗУ емкостью 32 кбайт 27C256, а для V-ROM – емкостью 8 кбайт 27C64 или K573PФ6.

Теперь необходимо на базе печатной платы от неисправного Dendy-картриджа смастерить схему, аналогичную простому SV-картриджу (рис.10), запаять две 28-контактные панельки под DD1 и DD2, установить в них прошитые микросхемы ПЗУ, выполнить монтаж тонким проводом и – самодельный картридж готов! Освоив детально программирование в среде процессора MC6502, можно в дальнейшем самостоятельно русифицировать Dendy-программы, вводить в них «бессмертие» или же оригинальные начальные заставки.

Литература

1. Рюмик С. Ремонт процессорной платы для «Dendy»/Радиолюбитель. Ваш компьютер.– 1997.–№4, С.22-26.
2. Веремеенко С. Видеопроектор для ZX-SPECTRUM/ZX-PEBIO.– 1995.–№6.–С.2-22.
3. Веремеенко С. Адаптация игр 8-битовых видеоприставок для ZX-SPECTRUM с видеопроцессором/ZX-PEBIO.–1996.– №4-5.–С.5-15.
4. Веремеенко С. DENDY под микроскопом/Радиолюбитель. Ваш компьютер.–1996.–№2.–С.24, 25; №3.–С.22-24; №4.–С.24-26.
5. Веремеенко С. Подробнее о видеопроцессоре DENDY/Радиолюбитель. Ваш компьютер.–1996.–№9.–С.25-27.
6. Насковец И., Ляхов В. Универсальный картридж для DENDY/Радиолюбитель. Ваш компьютер.–1997.–№11.–С.33, 34; №12.–С.31, 32.
7. Корчак А.Е. Язык программирования Бейсик для микро-ЭВМ. – М.: МЦНТИ, 1988.–131 с.
8. Программирование от А до Я '99. – CD-ROM, 1999.
9. Морер У. Язык ассемблера для персонального компьютера Apple/Пер. с англ. – М.: Мир, 1987.–430 с.