ТЕСТ ОЗУ 128К

ТЕСТ 128 v1.1 предназначен для проверки правильности

функционирования персональных компьютеров, совместимых с компью-

тером ZX-SPECTRUM-128, с помощью программных средств.

Для удобства использования тестовая программа записана в

микросхему ПЗУ.

ТЕСТ 128 v1.1 ВКЛЮЧАЕТ В СЕБЯ ПРОВЕРКУ

СЛЕДУЮЩИХ ЭЛЕМЕНТОВ КОМПЬЮТЕРА:

1. Проверка правильности функционирования цветоформирующего ка-

нала и регистра цветов бордюра.

2. Проверка правильности отображения на экране цветовых атрибу-

тов и соответствия адресов, формируемых видеоконтроллером, адре-

сам, формируемым процессором для области цветовых атрибутов эк-

рана, а также проверка правильности формирования видеоконтролле-

ром графической структуры экрана и соответствия адресов, форми-

руемых видеоконтроллером, адресам, формируемым процессором для

графической области экрана.

3. Проверка функционирования канала звука и выхода сигнала для

записи на магнитофон.

4. Проверка исправности буфера чтения данных из ОЗУ в процессор.

5. Проверка порта конфигурации.

6. Предварительная проверка исправности микросхем ОЗУ.

7. Обнаружение сбоев, сложных ошибок в микросхемах ОЗУ с помощ-

ью мощного псевдослучайного теста памяти, основанного на псев-

дослучайной последовательности.

ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ ТЕСТА 128 v1.1

1. Программа теста не использует стека и других ячеек ОЗУ

компьютера и может функционировать при полностью неисправном

ОЗУ.

2. Отображение результатов диагностики на экран производится

таким образом, что результаты могут быть прочитаны при од-

ной-двух полностью неисправных микросхем ОЗУ.

3. Все результаты диагностики выводятся на экран на русском

языке.

ТРЕБОВАНИЯ

Микросхема ТЕСТ-ПЗУ типа 573РФ2(5) устанавливается в па-

нельку так, чтобы 12 вывод ПЗУ 573РФ2(5) совпадал с 15 выводом

панельки. При этом необходимо отогнуть 21 и 24 выводы, соединить

их перемычкой и завести на +5V (28 вывод панельки). В случае ис-

пользования двух и более микросхем, ТЕСТ-ПЗУ устанавливается на

место младшей (0-й) микросхемы.

ВНИМАНИЕ!

ДЛЯ ПРАВИЛЬНОЙ РАБОТЫ ТЕСТА НЕОБХОДИМО, ЧТОБЫ ОТОГНУТЫЕ ВЫВОДЫ

НЕ КАСАЛИСЬ КОНТАКТОВ ПАНЕЛИ - В ПРОТИВНОМ СЛУЧАЕ ОТДЕЛЬНЫЕ ЭТА-

ПЫ ТЕСТА МОГУТ ВЫПОЛНЯТЬСЯ НЕПРАВИЛЬНО ИЛИ СОВСЕМ НЕ РАБОТАТЬ!!!

ОБ УСТРОЙСТВЕ ZX-SPECTRUM-128

Компьютер ZX-SPECTRUM-128 имеет ОЗУ объёмом 128К, которое

разделено на 8 частей по 16 Килобайт и которые условно называют-

ся банками памяти, а также имеют номера от 0 до 7. Так как про-

цессор Z80 имеет адресное пространство только 64К, то нет воз-

можности разом подключить к процессору всю оперативную память.

Выход был найден следующим образом: представим себе адресное

пространство микропроцессора Z80, т.е. 64К разбитым на 4 сектора

по 16К каждый и пронумеруем из от 0 до 3. К каждому из 4-х сек-

торов подключены следующие устройства: в 0-м секторе - ПЗУ сис-

темы (в начальном состоянии, после сброса, это прошивка BASIC

128, в дальнейшем она может быть программным образом заменена

прошивкой BASIC 48 для совместимости с моделью ZX-SPECTRUM-48).

В первом секторе находится один из 16-ти килобайтных банков ОЗУ,

часть которого является памятью дисплея и отображается в область

экрана - в начальном состоянии это 5-й банк памяти, но в процес-

се работы он может быть заменён на 7-й.

Во втором секторе жёстко находится 2-й банк ОЗУ. В третий

сектор может быть подключён любой из 8-и банков ОЗУ, после сбро-

са туда подключён 0-й банк.

В системе имеется порт конфигурации, который доступен прог-

раммисту и имеет адрес #7FFD. С помощью записи в этот порт (ре-

гистр) мы можем менять конфигурацию системы - т.е. переключать

ПЗУ, банки ОЗУ и т.д. Регистр конфигурации имеет 8 разрядов.

Первые три (с 0 по 2-й) разряда определяют один из 8-ми банков

ОЗУ, подключенный в последний сектор адресного пространства про-

цессора. Третий определяет один из 2-х банков, который будет

подключён в первый сектор адресного пространства и отображён на

экран, 0 - подключён 5-й банк, 1 - подключён 7-й банк, 4-й раз-

ряд определяет ПЗУ: 0 - подключена прошивка BASIC 128, 1 - BASIC

128. Пятый разряд заведует блокировкой записи в регистр конфигу-

рации: 0 - запись разрешена, 1 - запись блокирована. Это необхо-

димо для работы в режиме ZX-SPECTRUM-48.

АДРЕС Z80 СЕКТОР ПЗУ 32К

0000-3FFFF ╔═══════╗ ┌─── 0 в разр. 5 ──╔═══════════╗

║ 16 К ║ │ ║ BASIC 128 ║

║0 сект.║──┴─── 1 в разр. 5 ──╢ BASIC 48 ║

╚═══════╝ ╚═══════════╝

╔═══════╗ ОЗУ 128К

4000-7FFF ║ 16 К ║ ╔═══════════╗

║1 сект.║──┬─── 1 в разр. 4 ──║ 7-й банк ║

╚═══════╝ │ ╠═══════════╣

│ ║ 6-й банк ║

│ ╠═══════════╣

└─── 0 в разр. 4 ──║ 5-й банк ║

╠═══════════╣

╔═══════╗ ║ 4-й банк ║

8000-BFFF ║ 16 К ║ ╠═══════════╣

║2 сект.║────────┐ ║ 3-й банк ║

╚═══════╝ │ ╠═══════════╣

└────────────║ 2-й банк ║

╔═══════╗ ╠═══════════╣

C000-FFFF ║ 16 К ║ ║ 1-й банк ║

║3 сект.║──<- любой из 8-и ╠═══════════╣

╚═══════╝ банков ОЗУ ║ 0-й банк ║

╚═══════════╝

000 в разрядах 2-0 - 0-й банк

001 в разрядах 2-0 - 1-й банк

... ... ... ...

111 в разрядах 2-0 - 7-й банк

ПУСК ТЕСТА В РАБОТУ

Тест автоматически начинает свою работу при включении пита-

ния или после нажатия кнопки 'СБРОС' компьютера. При этом в те-

чении 1-2 секунд на бордюре появляются полосу всех цветов (от

чёрного до белого). Они должны либо стоять неподвижно, либо мед-

ленно перемещаться. Это полезно, во-первых, для проверки наличия

всех цветов на бордюре, а во-вторых, в качестве задержки между

этапами. Правда стоит заметить, что чёткие полосы мы будем ви-

деть не на всех вариантах ZX-Spectrum-128. Из-за различия в так-

товой частоте и реализации сигнала WAIT на некоторых машинах

вместо полос мельтешение цветов, хотя машины работали исправно.

Но во всяком случае на варианте, который создал Сергей Зонов на

базе своей улучшенной платы для ZX-Spectrum-48, цветные полосы

как приклеенные.

ТЕСТ АТРИБУТОВ И ГРАФИКИ:

Учитывая, что тест атрибутов (как и тест графики) необходим

только на начальном этапе отладки компьютера, мы, в отличии от

тестов SPECTRUM 48K, попадаем в него только в том случае, если

нажимаем клавишу SPACE/BREAK на клавиатуре компьютера перед

окончанием самой первой задержки. Причём клавиша также может

быть нажата и удерживаема заблаговременно.

Если компьютер находится на столь ранней стадии отладки,

что не имеет клавиатуры, тогда вместо нажатия клавиши <SPACE>

можно подать уровень логического 0 на вход порта клавиатуры D0

шины данных системы.

Если вы нажали клавишу <SPACE>, после задержки, экран

начнёт медленно очищаться от цветовых атрибутов. При этом на

смену хаотически мигающим цв. знакоместам будут приходить знако-

места чёрно-белые, нормальной яркости и не мигающие (при этом

зона цветовых атрибутов ОЗУ с адреса #5800 по #5AFF будет запи-

сываться код #38). Заполнение области атрибутов происходит мед-

ленно, давая возможность проследить стр-ру экрана. Атрибуты

должны переписываться начиная с левого верхнего левого угла сле-

ва направо и сверху вниз по строкам - так, как мы читаем книгу.

Одновременно по бордюру должны бежать красно-голубые полосы, а

на выходах сигнала 'ЗВУК' и 'МАГНИТОФОН' должен присутствовать

сигнал 'МЕАНДР' (импульсы со скважностью 2 и частотой около

800Гц).

После очистки цветовых атрибутов на экране мы увидим хаоти-

чески расположенные чёрные точки на белом фоне. Затем начнётся

медленное заполнение области экрана байтами #FF, что воспринима-

ется на глаз, как рисование чёрных линий. При правильной стр-ре

экрана линии должны появляться по той же схеме, как и при заг-

рузке картинки на экран с магнитофона, т.е. сначала рисуется ли-

ния шириной в одну точку в самой верхней строке экрана; следую-

щая рисуется на 8 точек ниже неё; следующая ещё на 8 точек ниже;

и так 8 линий. Затем 9-я линия рисуется во второй строке, деся-

тая на 8 точек ниже 9-й, и т.д., пока верхняя часть экрана не

окажется полностью закрашенной в чёрный цвет, после чего начина-

ет закрашиваться вторая часть экрана, а за ней и нижняя треть

экрана. В конце концов весь экран должен оказаться закрашенным

чёрным цветом. Весь этот процесс сопровождается чёрно-белыми по-

лосами на бордюре и сигналом частотой 800Гц на выходах 'ЗВУК' и

'МАГНИТОФОН'.

Этот тест позволяет выявит такие неисправности как 'залипа-

ния' или обрывы адресных шин, ведущих от мультиплексоров к мик-

росхемам ОЗУ и от процессора или видеоконтроллера к мультиплек-

сорам, а также неисправность или неправильное включение самих

мультиплексоров или микросхем ОЗУ и т.д.

Скажем, если одновременно рисуется не одна линия, а две или

более, то эта неисправность линии адреса, выдаваемого видео-

контроллером в районе средних адресов. Если линия одновременно

начинает рисоваться в нескольких местах одной и той же строки,

это означает неисправность линии адреса видеоконтроллера в райо-

не младших адресов (A0-A4) и т.д.

Т.е. путём анализа видимой на экране картины рисования ли-

ний можно составить представление о характере неисправности.

Помимо проверки стр-ры экрана мы можем убедиться в способ-

ности ячеек ОЗУ, содержимое которых отображается на экране, за-

писывать и хранить единицы и нули, что немаловажно для чита-

бельности сообщений в следующих частях теста.

ТЕСТ БУФЕРА ЧТЕНИЯ ДАННЫХ

Для проверки буфера чтения данных из ОЗУ программа теста

производит запись в первую ячейку ОЗУ (адрес #4000), а затем

считывает записанный байт и сравнивает его с тем, который запи-

сался. Проверка производится всеми возможными комбинациями от

#00 до #FF.

Если все комбинации считаны правильно, то на экран будет

выдано следующее сообщение:

ТЕСТ БУФЕРА

БУФЕР-НОРМА

а при ошибке: ТЕСТ БУФЕРА

ОШИБКА

ЗАПИСАНО - 00100000

СЧИТАНО - 00000000

Выдаются записанный и считанный байты в двоичном коде, что

позволяет быстро определить неисправность конкретного буфера

чтения данных. (При этом возможна как неисправность микросхемы

буфера, так и обрыв соответствующей линии данных, ведущей от бу-

фера к микросхеме ОЗУ или к процессору, а также полная нерабо-

тоспособность микросхемы ОЗУ соответствующего разряда шины дан-

ных. Однако в последнем случае эта неисправность может быть вы-

явлена простой перестановкой микросхем ОЗУ на плате и повторени-

ем теста.)

Если тест прошёл успешно, то после задержки программа пе-

рейдёт к тесту порта конфигурации, а если обнаружена ошибка, то

вышеуказанное сообщение будет сохранено на экране, причём бордюр

будет чёрным, а программа зависнет, так как, если неработоспосо-

бен буфер, то нет смысла продолжать тест.

ТЕСТ ПОРТА КОНФИГУРАЦИИ

Порт конфигурации, как говорилось выше, имеет адрес в сис-

теме #7FFD и предназначен для подключения различных банков памя-

ти в адресное пространство процессора. Во время теста мы записы-

ваем байты, отличающиеся на 1 в первую ячейку каждого из 8-и

банков, последовательно подключая их в старший сектор (для про-

цессора адрес первой ячейки любого подключенного банка - #C000).

Затем производится считывание этих байт и, если они совпали с

записанными, то байт записываемый в первую ячейку 0-го банка,

увеличивается на единицу и процесс повторяется до тех пор, пока

до нулевого банка не будут перебраны все комбинации от #00 до

#FF. Т.е. на первом проходе в 0-й банк пишется #00, в 1-й - #01,

во 2-й - #02,... , в 7-й #07.

На втором проходе в 0-й банк - #01, в 1-й - #02,... , в 7-й

- #08 и так далее до 255 прохода (#FF).

После этого порт конфигурации считается исправным и вы-

даётся сообщение:

ТЕСТ - ПОРТ КОНФИГУРАЦИИ

ПОРТ КОНФИГУРАЦИИ - НОРМА

Если обнаружено несовпадение считанного с записанным бай-

том, то выдаётся сообщение об ошибке:

ТЕСТ - ПОРТ КОНФИГУРАЦИИ

ОШИБКА

СЕКТОР - 3 ЗАПИСАНО - 00000000

БАНК - 0 СЧИТАНО - 00000010

Так следует обратить внимание на следующие моменты: чтобы

проверить правильность начальной установки 0-го банка в послед-

ний сектор адресного пространства тестирующая программа записи

его в первую ячейку байта 00 не производит записи в порт конфи-

гурации никаких значений, а считает, что после сброса 0-й банк

уже установлен. Поэтому, если программа не может считать байт 00

из 0-го банка, то причиной этого может оказаться неверная на-

чальная установка, хотя я не даю каких-либо возможных картин

возможных неисправностей, например такое же сообщение будет вы-

дано при постоянном 0 на выходе 1-го разряда порта конфигурации

или при замыкании выходов 0-го и 1-го разрядов.

При ошибке в данной части теста предстоит попытаться проа-

нализировать ситуацию и проверить схему на возможные неисправ-

ности. Например, в вышеприведённом примере следует попытатьтся

понять, почему при попытке чтения из 0-го банка мы получили

байт, который послали во 2-й банк. Ещё один момент: байты, запи-

санные в первые ячейки банков 2 и 5, мы пытаемся прочесть не

только из ячейки с номером #C000, но и из первых ячеек других

секторов. Для банка 2 - как по адресу C000, так и по адресу

8000; а для банка 5 - как по адресу C000, так и по адресу 4000.

При несовпадении записанного и считанного будет выведено сообще-

ние:

для банка 2 - СЕКТОР - 2 для банка 5 - СЕКТОР - 1

БАНК - 2 БАНК - 5

Это означает, что, видимо, вместо этих банков подключено

бог знает что. Естественно, чтобы тест прошёл успешно, необходи-

мо, чтобы первые ячейки ОЗУ были бы исправны, в противном случае

будет выдано сообщение об ошибке, хотя сам порт исправен. Выход

может быть найден перестановкой микросхем ОЗУ.

ПРЕДВАРИТЕЛЬНЫЙ ТЕСТ ОЗУ

Предварительный тест ОЗУ, несмотря на все его недостатки

(обнаружение не всех ошибок ОЗУ), проверяет адресные линии. Он

способен отличить ошибку адреса от ошибки данных, проверяет спо-

собность ячеек ОЗУ хранить '0' и '1', и главное, делает это

очень быстро. Это достигается записью на первом проходе всей об-

ласти ОЗУ кодовой последовательностью.

11011011 (#DB)

10110110 (#B6)

01101101 (#6D)

После записи всей последовательности происходит считывание

получившихся кодовых последовательностей. Например мы записыва-

ли:

АДРЕСА БАЙТЫ

... A2 A1 A0 HEX

---------------------------

... 0 0 0 DB

... 0 0 1 B6

... 0 1 0 6D

... 0 1 1 DB

... 1 0 0 B6

... 1 0 1 6D

... 1 1 0 DB

... 1 1 1 B6

Предположим, что у нас имелся обрыв линии адреса

'A2'(A2=0), тогда у нас получится следующая комбинация:

АДРЕСА БАЙТЫ

... A2 A1 A0 HEX

---------------------------

... 0 0 0 DB B6

... 0 0 1 B6 6D В эту область ОЗУ

... 0 1 0 6D DB процессор запишет

... 0 1 1 DB B6 контрольные коды дважды

... \* 0 0 \*\*

... \* 0 1 \*\* Эта область недоступна

... \* 1 0 \*\* процессору

... \* 1 1 \*\*

При такой неисправности программа обнаружит отказ линии и

выдаст сообщение типа:

ТЕСТ ОЗУ

ОШИБКА

АДРЕСА АДРЕС - C000

СЕКТОР - 3 ЗАПИСАНО - 11011011

БАНК - 0 СЧИТАНО - 10110110

Если же обнаружено несовпадение лишь одного бита в байте,

то это будет считаться отказом какой-либо из восьми микросхем

ОЗУ и будет выдано сообщение:

ТЕСТ ОЗУ

ОШИБКА

РАЗРЯДА 2 АДРЕС - D843

СЕКТОР - 3 ЗАПИСАНО - 11011011

БАНК - 6 СЧИТАНО - 11011111

Что означает следующее: во время теста ОЗУ выявлена ошибка

микросхемы ОЗУ, соответствующая шине данных D2 процессора Z80.

Запись, а затем и чтение производилось из последней четверти ад-

ресного пространства процессора (из 3-го сектора с адреса C000

по FFFF; следует заметить, что как в предварительном, так и в

бесконечном тестах ОЗУ эта информация служит скорее напоминани-

ем, по той причине, что чтение-запись производится только из

этого сектора т.е. 3). С помощью порта конфигурации в этот сек-

тор подключён 6-й и 16-и килобайтный банк из имеющихся с 0-го по

7-й (128К). Конкретный адрес ошибки - D843, по этому адресу был

записан 0, а считана 1.

Второй проход отличается от первого только другой последо-

вательностью записываемых в ОЗУ байт:

00100100 (#24)

01001001 (#49) Выводимые при этом сообщения те же.

10010010 (#92)

Следует обратить внимание ещё на один момент. Во время опе-

раций записи и чтения, как в предварительном, так и в бесконеч-

ном тесте ОЗУ цвет бордюра даёт нам информацию о том, какой из

банков подключен - чёрный цвет - подключён 0-й банк, синий - 1-й

и т.д. Байт, который программа пишет в порт конфигурации, дубли-

руется в порте цветов бордюра. Во время теста ОЗУ можно видеть

как происходит заполнение 128к, а затем считывание; при этом в

обоих случаях цвет бордюра меняется от чёрного до белого.

Если же ОЗУ исправно, на экран выводится сообщение:

ТЕСТ ОЗУ

ОЗУ - НОРМА

После этого начинает выполняться мощный бесконечный тест

ОЗУ. Если найдена ошибка, то тест зависает и дальше не идёт.

Часто приходится сталкиваться с такой ситуацией: некто тес-

тировал ОЗУ в чужом компьютере и там оно работает хорошо, из

этого человек делает вывод, что оно обязательно должно также хо-

рошо работать и в его машине. А тест может выдавать ошибки то в

одной микросхеме ОЗУ, то в другой, то в третьей. Значит, делает

вывод неудачный творитель компьютера, дело в тесте, он работает

неправильно. Очевидно, тут дело, как правило, не в самих микрос-

хемах ОЗУ, а в плохом питании, отсутствии шунтирующих конденса-

торов, 'натянутых' временных диаграммах сигналов управления ОЗУ

(WR, RAS, CAS и адресных мультиплексоров). Поэтому, настраивая

плату со сбоями ОЗУ, следует попробовать следующее:

1. Более тщательно посмотреть монтаж, промыть плату от ка-

нифоли, усилить и пропаять подозрительные печатные проводники в

районе установки микросхем ОЗУ, а также микросхем, вырабатываю-

щих управляющие сигналы и синхронизацию.

2. Поставить шунтирующие конденсаторы рядом с микросхемами

ОЗУ (как керамику так и электролиты).

3. Применить более быстродействующие ОЗУ (с меньшей буквой)

- если стояла Д - надо поставить Г, если Г - то В и т.д.

4. Улучшить сигнал CLC процессора путём уменьшения резисто-

ра между этим сигналом и +5V.

5. Попытаться, если в вашем типе компьютера есть подозрения

на то, что временные диаграммы 'натянуты', ускорить некоторые из

сигналов управления (резистор между сигналом и +5V), либо наобо-

рот, 'затянуть' резистор или конденсатор на 'землю'. Иногда, ес-

ли один и тот же сигнал используется как в качестве сигнала RAS,

так и в качестве сигнала управления мультиплексорами, то, чтобы

затянуть один из этих сигналов, оставив другой неизменным, можно

'врезать' резистор порядка 50-100 Ом и только после него ставить

конденсатор.

МОЩНЫЙ БЕСКОНЕЧНЫЙ ТЕСТ ОЗУ

Бесконечный тест ОЗУ предназначен как для выявления сбоев в

течении длительного времени, так и для выявления сложных ошибок,

которые не смог обнаружить предварительный тест. Бесконечный

тест генерирует псевдослучайную последовательность и пишет её в

память. Он будет бесконечно тестировать ОЗУ до тех пор, пока не

будет обнаружена ошибка. После этого будет выдана такая же диаг-

ностика, как и при предварительном тестировании ОЗУ. Рекоменду-

ется при обнаружении ошибки запускать тест несколько раз кнопкой

<СБРОС>. Если тест обнаруживает сбой не в одной, а в нескольких

(или всех микросхемах), то не спешите менять ОЗУ, а воспользуй-

тесь рекомендациями, приведёнными выше - видимо ОЗУ не виновато.