

Тестер микросхем

AVC1READER ver.1

Руководство пользователя

Содержание

Описание	2
1. Порядок работы с тестером AVC1READER	4
1.1 Включение	4
1.2 Подключение сигнальных шлейфов	5
1.3 Запись AVC файла на SD карту	11
1.4 Подключение SD карты.....	13
1.5 Запуск тестирования и поиск сигналов.....	14
1.6 Процесс тестирования.....	15
1.7 Получение результатов	16
2. Отладочная индикация.....	17
Заключение.....	19

Описание

AVC1READER представляет собой устройство-тестер цифровых микросхем, блоков и модулей, в частности, синтезированных в ПЛИС, построенный на базе отладочного комплекта Terasic DE2-115. Логический проект тестера написан для ПЛИС EP4CE115F29 фирмы Altera семейства Cyclone IV, которая входит в состав данного отладочного комплекта и сохранен на загрузочной флеш-памяти EPCS64, которая также присутствует на плате комплекта.

Принцип работы с тестером AVC1READER заключается в следующем. Плата с испытуемой микросхемой подключается с помощью 40- (без экранирования) или 80- (с экранированием) жильных шлейфов с защитными резисторами к колодке выводов 2x20 (разъем IDC-40) на плате DE2-115 (1) и к колодкам на плате расширения THDB-HTG. **Подключать обычный шлейф без защитных резисторов запрещено.** В сумме, на всех колодках, могут быть задействованы до 108 сигналов. Настраиваться каждый сигнал может либо на ввод, либо на вывод, либо может быть двунаправленным. Максимальное количество выводов в проекте для ПЛИС установлено как 128, но реально из них используется только 104.

После подключения шлейфов, включается AVC1READER с помощью кнопки, индикация загорается. Далее вставляется SD карта, предварительно подготовленная и отформатированная в файловую систему FAT32 с помощью скрипта AVC1WRITER.sh.

Для взаимодействия тестера AVC1READER с пользователем, используются устройства ввода-вывода, предусмотренные на плате DE2-115, такие как знакосинтезирующий ЖК дисплей, светодиоды, 7-сегментные индикаторы, переключатели и кнопки.

После установки SD карты устройство-тестер сбрасывается посредством нажатия на кнопку RESET на плате DE2-115, и на знакосинтезирующем ЖК дисплее необходимо дождаться информации о наличии или отсутствии AVC файла на SD карте. Важно, чтобы файл на SD карте имел расширение *.avc, иначе он не будет обнаружен тестером. В случае если SD карты нет или она читается с ошибками, после сброса, информация о AVC файле не появляется, а загорается надпись "No SD Card". В это же время, информация на 7-сегментных индикаторах отображает начальное состояние и готовность AVC1READER начинать тестирование микросхемы.

Светодиоды на плате DE2-115 служат только для отображения внутренних состояний машин состояний AVC1READER. Исключением является светодиод готовности, который сигнализирует об окончании тестирования или остановке теста, к примеру, из-за ошибки.

Чтобы начать тестирование микросхемы с помощью AVC1READER, необходимо убедиться, что выводы на колодках, идущих к испытуемой микросхеме, скоммутированы верно и нигде выход AVC1READER не

замкнут на выход тестируемой микросхемы. Коммутация проверяется по порядку следования сигналов в AVC файле.

После нажатия на кнопку START, тестер AVC1READER начинает процедуру тестирования и, в первую очередь, самостоятельно, следуя по AVC файлу, находит сигналы и определяет их тип (выход, вход или двунаправленный). Важно, чтобы в AVC файле присутствовали двунаправленные или входные сигналы, которые имеют обозначение “х”, “z”, “H” или “L” в векторах. Если тестер не обнаружит таковых сигналов, ему будет нечего тестировать, так как все сигналы окажутся для него выходными, и он выдаст ошибку на дисплее “No 3-state signl”. Поиск сигналов может занимать довольно большое количество времени, так как производится прямой последовательный перебор сигналов в векторах AVC файла, загруженного на SD карту. Поэтому время поиска напрямую зависит от размера загружаемого AVC файла.

Если во время тестирования были найдены двунаправленные сигналы, AVC1READER начинает процедуру чтения AVC файла и поиска ошибок. Ошибки представляют собой результат сравнения двунаправленных или входных сигналов типа “х”, “z”, “H” или “L” с сигналами, следующими из тестируемой микросхемы. При обнаружении несовпадения, ошибка фиксируется и записывается в ERCY файл в виде вектора.

Полученный ERCY файл с ошибками записывается на SD карту с именем E000000*.erc, где * - порядковый символ по таблице ASCII от 1 (0x31), который каждый новый тест увеличивается на единицу. Каждый новый тест (RESET, затем START), при наличии ошибок, перезаписывает текущий ERCY файл и присваивает ему следующий порядковый символ в его имени. Как правило, для конкретного AVC файла, без внесения в него изменений, достаточно провести всего 1 тест, либо 2, если возникают сомнения в правильности работы AVC1READER.

В случае если AVC файл подлежит изменению или должен быть заменен на другой, следует записать его заново с помощью скрипта AVC1WRITER.sh, который форматирует SD карту и загружает требуемый подготовленный AVC файл.

На SD карту можно записывать только 1 единственный AVC файл, иначе AVC1READER будет произведено чтение первого попавшегося AVC файла.

Вместе с AVC файлом на SD карту можно записывать другие файлы, но их форматы должны быть отличны от форматов AVC и ERCY.

Удалять без форматирования вручную и заново записывать, либо изменять AVC файлы на SD карте напрямую, в данной версии AVC1READER, нельзя, так как результат в ERCY может быть неверным. Записывать AVC файл можно вручную, без применения AVC1WRITER.sh можно, но тогда придется отформатировать SD карту в FAT32 и, прежде чем записывать AVC файл, необходимо добавить в него 1 последний вектор, в котором “R1 сус” заменено на “Xx XXX”, а все сигналы заменены на символ “х”. То есть “H”, “L”, “0”, “1” и “z” заменяются на “х”.

Записывать AVC файлы, размер которых превышает 4 Гб, нельзя, так как файловая система FAT32 не поддерживает файлы больше 4 Гб.

1. Порядок работы с тестером AVC1READER

1.1 Включение

Для начала работы с тестером AVC1READER, необходимо его включить. Для этого необходимо подключить блок питания 12 В 2 А к разъему (Рисунок 1, 1) и нажать красную кнопку (Рисунок 1, 2) до щелчка с фиксацией её положения. После включения необходимо немного подождать пока проект из конфигурационной памяти загрузится в ПЛИС (около секунды). После этого на дисплее (Рисунок 1, 3) должна появиться информация, а на каждом из 8-ми 7-сегментных индикаторов (Рисунок 1, 4) должно загореться по одному сегменту.

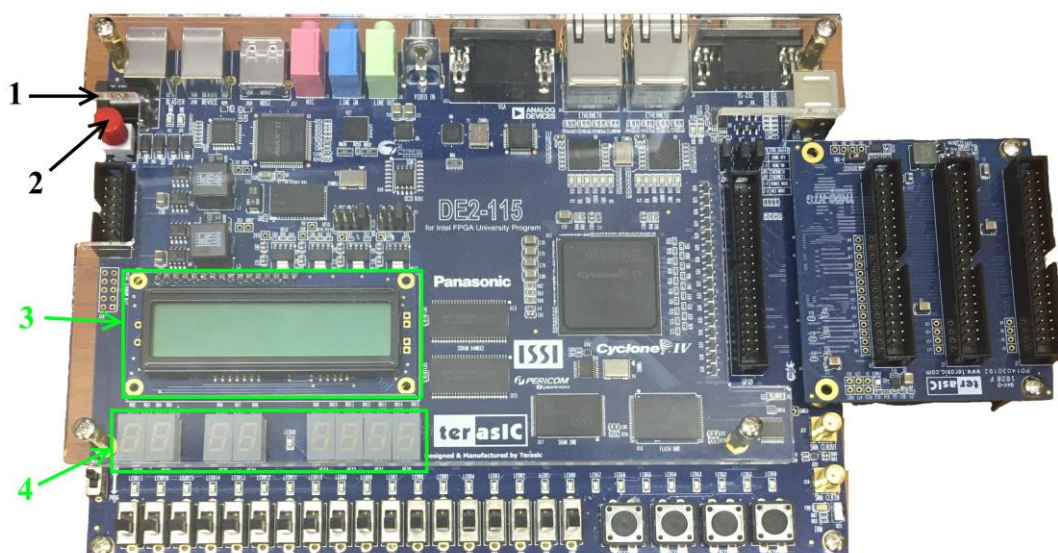


Рисунок 1. Поддача питания и включение тестера

Если плата расширения THDB-NTG (Рисунок 3, 1) подключена, то должны загореться светодиоды присутствия платы расширения D40 (Рисунок 3, 2) и питания платы расширения POWER (Рисунок 3, 3). Также должен загореться светодиод питания платы D1 (Рисунок 3, 4).

Плата расширения подключается к разъему HSMC, который изображен на рисунке 2.



Рисунок 2. Разъем HSMC

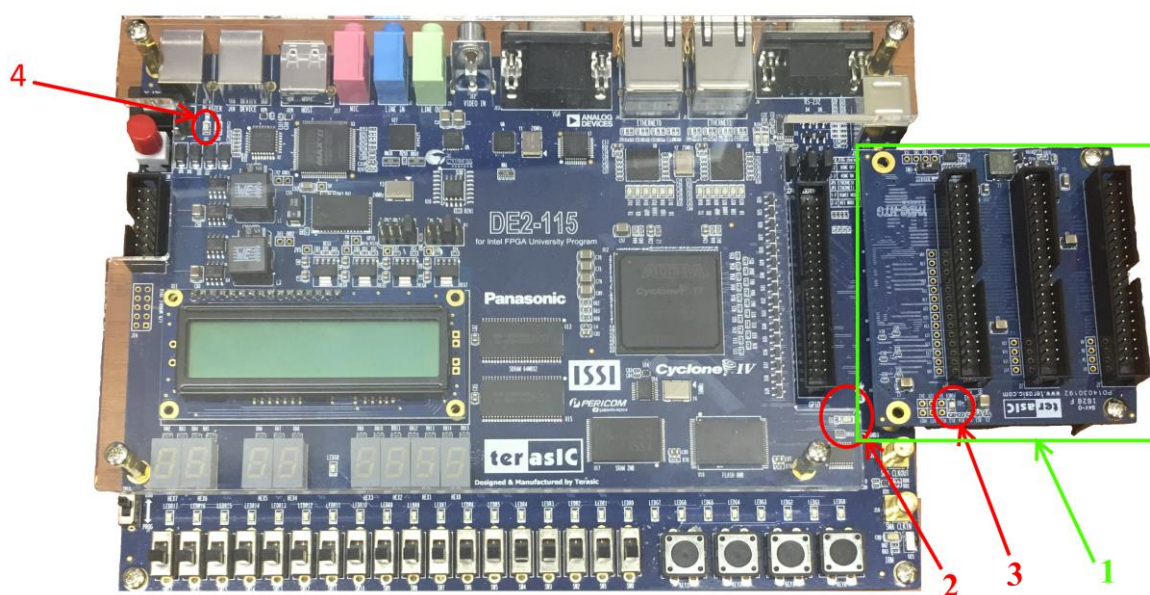


Рисунок 3. Плата расширения THDB-HTG и индикация

После того как индикация загорелась и всё работает верно, можно начинать подключать сигнальные шлейфы к разъемам IDC-40.

1.2 Подключение сигнальных шлейфов

Для проверки микросхемы тестером AVC1READER, необходимо подключить 40-жильный сигнальный шлейф с защитными резисторами (Рисунок 4) к соответствующим выводам отладочной платы с микросхемой и платы AVC1READER разъема IDC-40 2x20. Разъем имеет ключ (Рисунок 4), поэтому ориентироваться при подключении нужно по нему. **Подключать обычный шлейф без защитных резисторов запрещено.**



Рисунок 4. Штыревой разъем (слева) и подключаемый сигнальный шлейф с защитными резисторами (справа)

На отладочной плате DE2-115 имеется одна колодка IDC-40 (Рисунок 5, 1) для подключения шлейфа. Плата расширения THDB-HTG добавляет ещё 3 такие колодки (Рисунок 5, 2). Таким образом, доступно всего 4 колодки, что в сумме даёт максимально возможное количество анализируемых сигналов, равное 104.

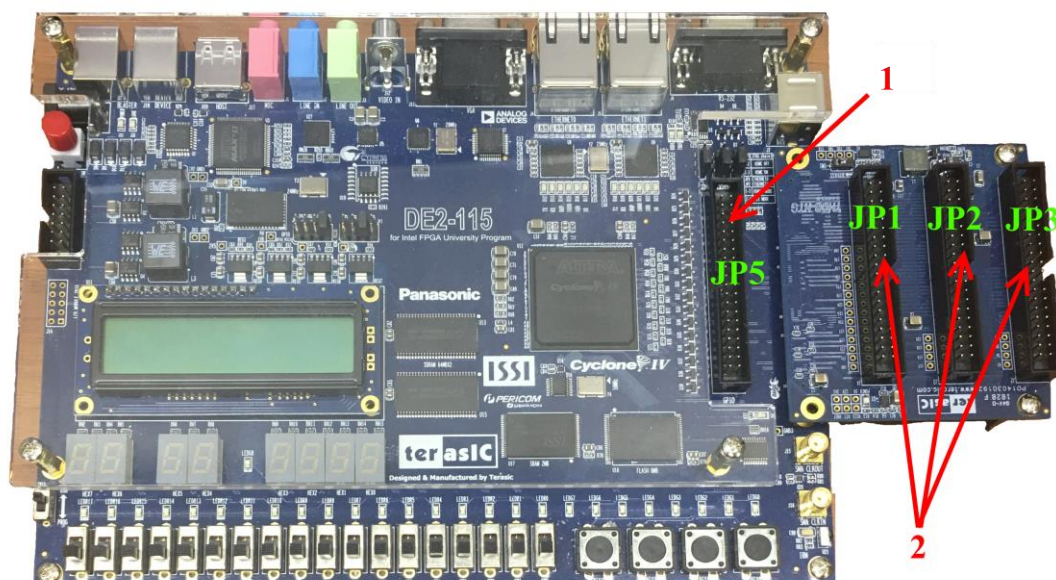


Рисунок 5. Колодки для подключения сигнальных шлейфов

Распиновки сигналов на колодках тестера и платы расширения представлены на рисунке 7. Черным прямоугольником помечен ключ, который представляет собой соответствующий вырез на разъеме IDC-40 2x20. Каждый сигнал из AVC файла, включая сигналы шин, соответствует определенному сигналу на колодке. Это соответствие иллюстрирует пример на рисунке 6, где изображены только 15 сигналов, включая сигналы с шины.

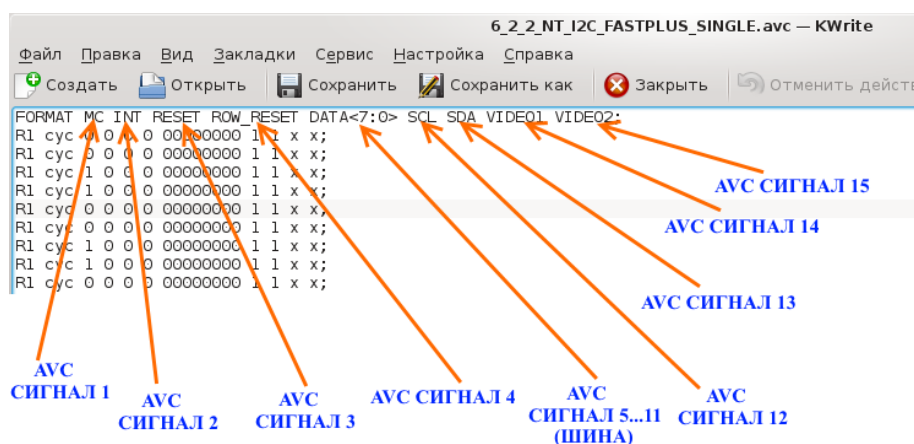


Рисунок 6. Распиновка колодок IDC-40 на платах DE2-115 и THDB-HTG

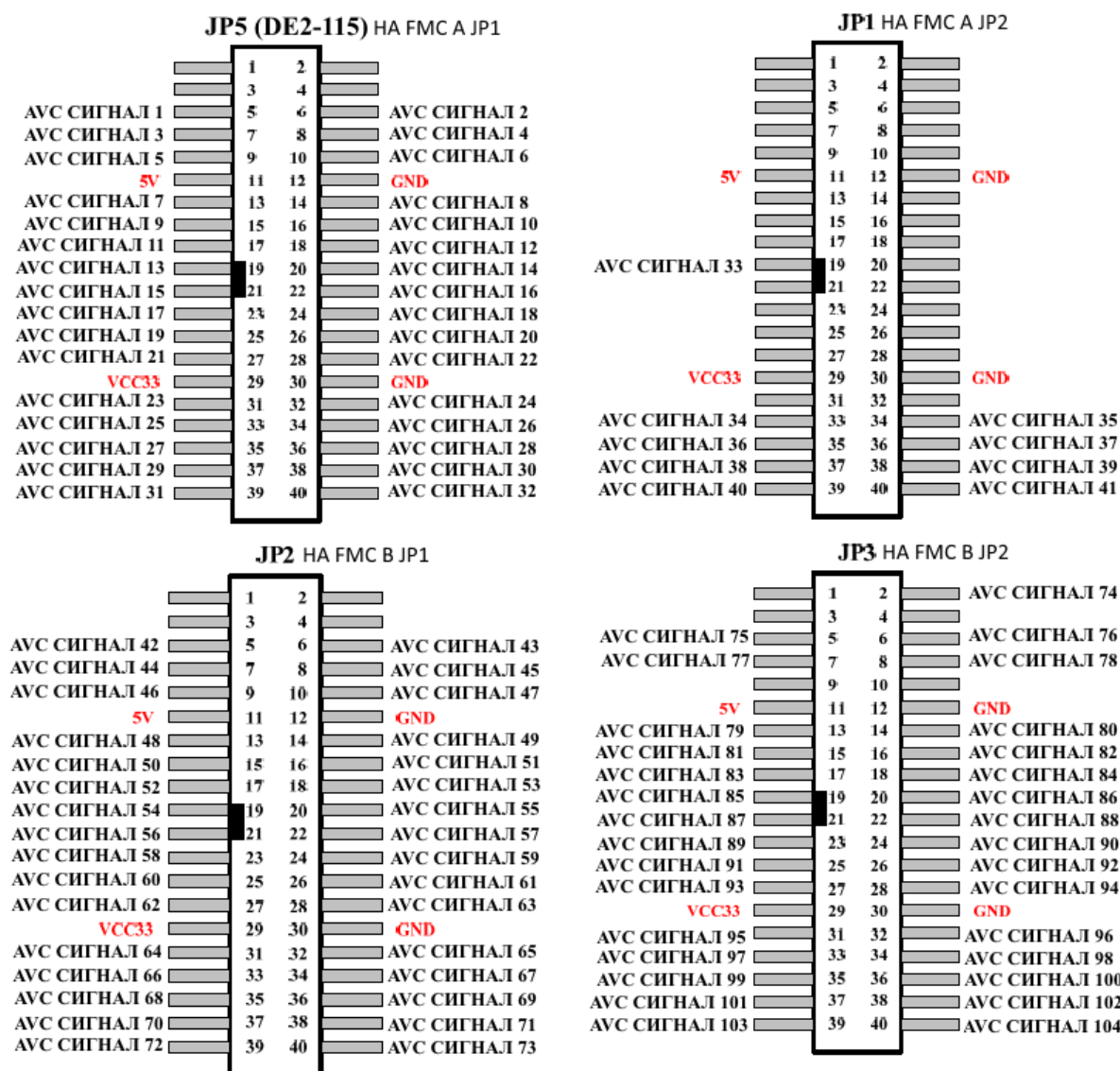


Рисунок 7. Распиновки колодок IDC-40 на платах DE2-115 и THDB-HTG

AVC сигналы с разъемов IDC-40 необходимо подключать к разъемам отладочной платы с тестируемой микросхемой таким образом, чтобы эти сигналы приходили на необходимые для теста выводы микросхемы. Для этого необходимо смотреть документацию на плату с микросхемой или ПЛИС с синтезируемым проектом микросхемы.

Чтобы было проще назначать выводы в проекте для платы Arria 10 GX FPGA Development Kit для ПЛИС Arria 10, подключенные к колодкам IDC-40 переходника F2G (Рисунок 8), далее приведем таблицу подключения выводов (Таблица 1). В таблице приводится соответствие выводов для колодок GPIO переходников F2G на разъемах FMC A и FMC B и сигналов из файла AVC. Из всего набора разъемов на переходнике F2G используются только JP1 и JP2, так как на плате Arria 10 GX используется только малый набор сигналов FMC, поэтому на остальные колодки вывода от ПЛИС просто не выведены.

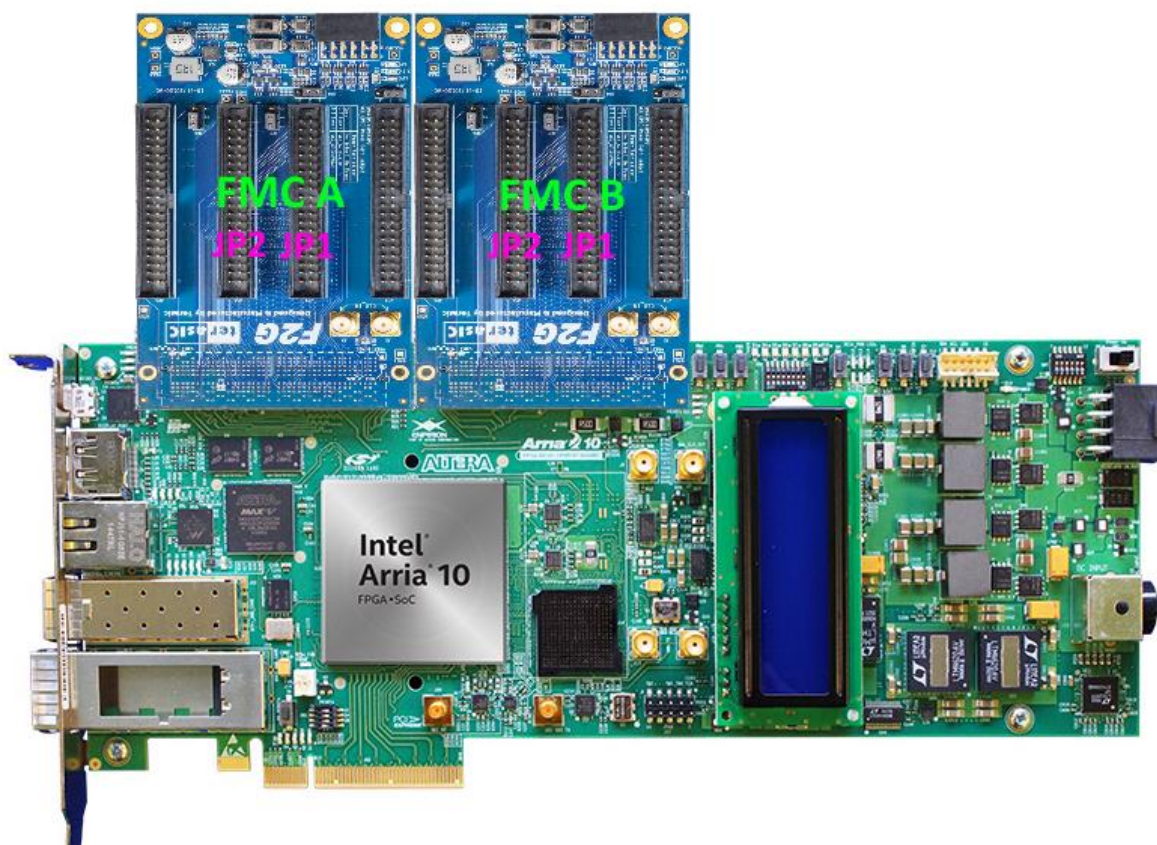


Рисунок 8. Модули F2G (FMC to GPIO) для отладочной платы Arria 10 GX
FPGA Development Kit

Таблица 1. Соответствие сигналов AVC файла контактам ПЛИС

Выводы ПЛИС на FMC	AVC сигналы
PIN_AT14	AVC СИГНАЛ 1 (FMC A JP1)
PIN_AR14	AVC СИГНАЛ 2 (FMC A JP1)
PIN_AR9	AVC СИГНАЛ 3 (FMC A JP1)
PIN_AT9	AVC СИГНАЛ 4 (FMC A JP1)
PIN_AW13	AVC СИГНАЛ 5 (FMC A JP1)
PIN_AV13	AVC СИГНАЛ 6 (FMC A JP1)
PIN_AR17	AVC СИГНАЛ 7 (FMC A JP1)
PIN_AP17	AVC СИГНАЛ 8 (FMC A JP1)
PIN_AV19	AVC СИГНАЛ 9 (FMC A JP1)
PIN_AW19	AVC СИГНАЛ 10 (FMC A JP1)
PIN_AU11	AVC СИГНАЛ 11 (FMC A JP1)
PIN_AU12	AVC СИГНАЛ 12 (FMC A JP1)
PIN_AY10	AVC СИГНАЛ 13 (FMC A JP1)
PIN_AY11	AVC СИГНАЛ 14 (FMC A JP1)
PIN_AU18	AVC СИГНАЛ 15 (FMC A JP1)
PIN_AT18	AVC СИГНАЛ 16 (FMC A JP1)
PIN_AT17	AVC СИГНАЛ 17 (FMC A JP1)
PIN_AU17	AVC СИГНАЛ 18 (FMC A JP1)

PIN_AN20	AVC СИГНАЛ 19 (FMC A JP1)
PIN_AP19	AVC СИГНАЛ 20 (FMC A JP1)
PIN_AR22	AVC СИГНАЛ 21 (FMC A JP1)
PIN_AT22	AVC СИГНАЛ 22 (FMC A JP1)
PIN_AV11	AVC СИГНАЛ 23 (FMC A JP1)
PIN_AW11	AVC СИГНАЛ 24 (FMC A JP1)
PIN_BB15	AVC СИГНАЛ 25 (FMC A JP1)
PIN_BC15	AVC СИГНАЛ 26 (FMC A JP1)
PIN_AT19	AVC СИГНАЛ 27 (FMC A JP1)
PIN_AT20	AVC СИГНАЛ 28 (FMC A JP1)
PIN_AY16	AVC СИГНАЛ 29 (FMC A JP1)
PIN_AW16	AVC СИГНАЛ 30 (FMC A JP1)
PIN_BC18	AVC СИГНАЛ 31 (FMC A JP1)
PIN_BD18	AVC СИГНАЛ 32 (FMC A JP1)
PIN_AW18	AVC СИГНАЛ 33 (FMC A JP2)
PIN_BA15	AVC СИГНАЛ 34 (FMC A JP2)
PIN_BA14	AVC СИГНАЛ 35 (FMC A JP2)
PIN_BB17	AVC СИГНАЛ 36 (FMC A JP2)
PIN_BB18	AVC СИГНАЛ 37 (FMC A JP2)
PIN_AY17	AVC СИГНАЛ 38 (FMC A JP2)
PIN_AW17	AVC СИГНАЛ 39 (FMC A JP2)
PIN_AV20	AVC СИГНАЛ 40 (FMC A JP2)
PIN_AU20	AVC СИГНАЛ 41 (FMC A JP2)
PIN_E11	AVC СИГНАЛ 42 (FMC B JP1)
PIN_E10	AVC СИГНАЛ 43 (FMC B JP1)
PIN_G10	AVC СИГНАЛ 44 (FMC B JP1)
PIN_F10	AVC СИГНАЛ 45 (FMC B JP1)
PIN_A15	AVC СИГНАЛ 46 (FMC B JP1)
PIN_A14	AVC СИГНАЛ 47 (FMC B JP1)
PIN_B12	AVC СИГНАЛ 48 (FMC B JP1)
PIN_A12	AVC СИГНАЛ 49 (FMC B JP1)
PIN_F13	AVC СИГНАЛ 50 (FMC B JP1)
PIN_G13	AVC СИГНАЛ 51 (FMC B JP1)
PIN_H11	AVC СИГНАЛ 52 (FMC B JP1)
PIN_H10	AVC СИГНАЛ 53 (FMC B JP1)
PIN_K16	AVC СИГНАЛ 54 (FMC B JP1)
PIN_K17	AVC СИГНАЛ 55 (FMC B JP1)
PIN_H13	AVC СИГНАЛ 56 (FMC B JP1)
PIN_J13	AVC СИГНАЛ 57 (FMC B JP1)
PIN_D13	AVC СИГНАЛ 58 (FMC B JP1)
PIN_C13	AVC СИГНАЛ 59 (FMC B JP1)
PIN_C15	AVC СИГНАЛ 60 (FMC B JP1)
PIN_B15	AVC СИГНАЛ 61 (FMC B JP1)

PIN_A17	АВС СИГНАЛ 62 (FMC B JP1)
PIN_B17	АВС СИГНАЛ 63 (FMC B JP1)
PIN_J19	АВС СИГНАЛ 64 (FMC B JP1)
PIN_K19	АВС СИГНАЛ 65 (FMC B JP1)
PIN_M13	АВС СИГНАЛ 66 (FMC B JP1)
PIN_L14	АВС СИГНАЛ 67 (FMC B JP1)
PIN_M15	АВС СИГНАЛ 68 (FMC B JP1)
PIN_L15	АВС СИГНАЛ 69 (FMC B JP1)
PIN_K20	АВС СИГНАЛ 70 (FMC B JP1)
PIN_L19	АВС СИГНАЛ 71 (FMC B JP1)
PIN_L20	АВС СИГНАЛ 72 (FMC B JP1)
PIN_K21	АВС СИГНАЛ 73 (FMC B JP1)
PIN_D11	АВС СИГНАЛ 74 (FMC B JP2)
PIN_A13	АВС СИГНАЛ 75 (FMC B JP2)
PIN_B13	АВС СИГНАЛ 76 (FMC B JP2)
PIN_D17	АВС СИГНАЛ 77 (FMC B JP2)
PIN_D16	АВС СИГНАЛ 78 (FMC B JP2)
PIN_G21	АВС СИГНАЛ 79 (FMC B JP2)
PIN_H21	АВС СИГНАЛ 80 (FMC B JP2)
PIN_A18	АВС СИГНАЛ 81 (FMC B JP2)
PIN_B18	АВС СИГНАЛ 82 (FMC B JP2)
PIN_E12	АВС СИГНАЛ 83 (FMC B JP2)
PIN_F12	АВС СИГНАЛ 84 (FMC B JP2)
PIN_B16	АВС СИГНАЛ 85 (FMC B JP2)
PIN_C16	АВС СИГНАЛ 86 (FMC B JP2)
PIN_H20	АВС СИГНАЛ 87 (FMC B JP2)
PIN_G20	АВС СИГНАЛ 88 (FMC B JP2)
PIN_G11	АВС СИГНАЛ 89 (FMC B JP2)
PIN_G12	АВС СИГНАЛ 90 (FMC B JP2)
PIN_H19	АВС СИГНАЛ 91 (FMC B JP2)
PIN_H18	АВС СИГНАЛ 92 (FMC B JP2)
PIN_M12	АВС СИГНАЛ 93 (FMC B JP2)
PIN_L13	АВС СИГНАЛ 94 (FMC B JP2)
PIN_K14	АВС СИГНАЛ 95 (FMC B JP2)
PIN_J14	АВС СИГНАЛ 96 (FMC B JP2)
PIN_M16	АВС СИГНАЛ 97 (FMC B JP2)
PIN_M17	АВС СИГНАЛ 98 (FMC B JP2)
PIN_L18	АВС СИГНАЛ 99 (FMC B JP2)
PIN_M18	АВС СИГНАЛ 100 (FMC B JP2)
PIN_M21	АВС СИГНАЛ 101 (FMC B JP2)
PIN_M20	АВС СИГНАЛ 102 (FMC B JP2)
PIN_J22	АВС СИГНАЛ 103 (FMC B JP2)
PIN_J21	АВС СИГНАЛ 104 (FMC B JP2)

1.3 Запись AVC файла на SD карту

В данной версии AVC1READER записывать AVC файл на SD карту вместе с другими файлами на ней можно, если до этого, после форматирования, на SD карте не было никаких AVC файлов. Результат в ERCY будет получен верный, однако файлы, находящиеся вместе с AVC файлом (кроме файла ERCY), могут пострадать, и первоначальная информация на них может быть утеряна. Поэтому, в данной версии AVC1READER, правильнее записывать AVC файл с помощью скрипта AVC1WRITER.sh. Скрипт AVC1WRITER.sh формирует конец файла для тестера AVC1READER, форматирует SD карту для дальнейшей записи на неё необходимого пользователю AVC файла и предотвращает пользователя от записи абсолютно пустого или ошибочного AVC файла, к примеру, с неверным заголовком.

Итак, чтобы записать AVC файл на SD карту правильно, необходимо придерживаться следующей последовательности действий:

1. Вставьте SD карту в соответствующий слот на ПК или через подключите SD карту с помощью кардридера;
2. Убедитесь, что скрипт AVC1WRITER.sh находится в одной директории с необходимым для записи AVC файлом;
3. Запустите скрипт из консоли Linux в виде *“./AVC1WRITER.sh <имя AVC файла>”*;
4. Если ошибок на этом этапе нет, то в консоли должно появиться сообщение *“Info: Opening and finalize AVC File. Please Wait ...”*. Если AVC файл большого размера, необходимо немного подождать;
5. Если имя AVC файла при запуске не задано, то в консоли будет выведено сообщение об ошибке *“Error: No filename in argument. Use AVCWRITER.sh <filename>”*. Укажите имя файла;
6. Если AVC файл не имеет расширения *.avc, либо если не указали это расширение в имени файла, то будет выведено сообщение об ошибке *“Error: AVC File <имя AVC файла> doesn't have an *.avc extention. Check for extention in <filename> field.”* Укажите расширение *.avc, которое обязательно для тестера;
7. Если файла не существует, то будет выведено сообщение об ошибке *“Error: AVC File <имя AVC файла> doesn't exists. Check <filename> field.”* Убедитесь, что файл с данным именем и расширением существует;
8. Если AVC файл пуст или заголовок в файле имеет какой-либо неверный формат, то будет выведено сообщение об ошибке *“Error: AVC File <имя AVC файла> doesn't have correct signal format for AVC1READER. Check AVC File for errors”*. Исправьте заголовок в AVC файле;
9. Если в AVC файле отсутствуют какие либо файлы в заголовке, то будет выведена ошибка *“Error: AVC File <имя AVC файла> doesn't have any signals. Check AVC File for at least 1 signal in header.”*. Добавьте сигналы в заголовок в AVC файле;

10. Если в AVC файле присутствуют вектора неверного формата, то в консоли будет выведена ошибка *"Error: AVC File <имя AVC файла> doesn't have correct vectors format for AVCIREADER. Check AVC File for errors."*. Исправьте ошибки в векторах в AVC файле.
11. В случае если файл пустой или в нем нет векторов, то в консоль будет выведено предупреждение: *"Warning: AVC File <имя AVC файла> doesn't have any vectors. Check AVC File for at least 1 vector."*
12. Далее будет предложено выбрать устройство, форматированное в FAT32, в частности, SD карта, а в консоли будет выведено *"Choose SD Device (number):"*;
13. Если устройств высветилось много, то, чтобы узнать, какое устройство является SD картой, необходимо отключить SD карту, либо кардридер, запустить скрипт и посмотреть какое из устройств отсутствует в списке. Отсутствующее устройство будет нашей SD картой;
14. Далее, в меню выбора SD карты *"Choose SD Device (number):"*, необходимо указать ту цифру, рядом с которой находится наша SD карта и нажать ENTER;
15. В случае если ни одного устройства с файловой системой FAT32 не найдено, в консоли появится сообщение *"Error: No SD Card or another device in system FAT32 formatted. Insert SD Card and PRESS ENTER"*. Необходимо вставить SD карту и нажать ENTER. Сообщение об ошибке должно пропасть, чтобы можно было продолжить;
16. После выбора SD карты, возможно, будет высвечено *"umount failed: Операция не позволена"*. Это сообщение не является ошибкой. Если карта не смонтирована, размонтирование лишнее, поэтому сообщение можно игнорировать;
17. Далее идет форматирование ранее выбранной SD карты в FAT32. В консоли при этом высвечивается сообщение *"Format <устройство> to FAT32 ... Please wait ..."*;
18. Если форматирование прошло успешно, то в консоли должно появиться сообщение *"Format done!"*;
19. Если форматирование по какой-либо причине не прошло, то будет выведено сообщение об ошибке *"Error: SD format not done. Maybe you have opened files on SD card."*. Одной из причин ошибки могут быть файлы на SD карте, которые в данный момент открыты или используются какими-либо программами;
20. Когда форматирование прошло, появится следующее сообщение *"Info: Now Mount or Reconnect and Open directory your SD card. And PRESS ENTER."*. Это означает, что необходимо монтировать (если есть права суперпользователя) либо переподключить SD карту и открыть её корневую директорию (Открыть вкладку *"Сменный носитель"* в системе) и после этого нажать ENTER;

21. Если SD карта не смонтирована, то после нажатия ENTER будет выведено сообщение: *"Error: SD Card not mounted. Try to open SD card directory and PRESS ENTER"*. Чтобы исправить ошибку и записать AVC файл, необходимо чтобы директория SD карты открылась;
22. Если SD карта смонтируется, далее будет произведено копирование AVC файла, ранее указанного при запуске скрипта. Выведется сообщение в консоль *"SD card mounted ! Trying to copy AVC file on SD card ... Please Wait ..."*;
23. После успешной записи AVC файла на SD карту будет выведено сообщение *"Info: Writing done! Now you can eject your SD card from slot and put it into AVC1READER slot !"*;
24. На том запись AVC файла заканчивается и можно проверить наличие файла на SD карте.

На этом запись файла с помощью скрипта заканчивается. Можно вынимать SD карту из слота или кардридера и подключать её к тестеру AVC1READER.

1.4 Подключение SD карты

Тестер микросхем AVC1READER тестирует микросхему согласно векторам, записанным в AVC файле. AVC файл считывается с SD карты, которая вставляется в специальный слот для SD карт, смонтированный на плате DE2-115 с обратной стороны (Рисунок 9, 1).

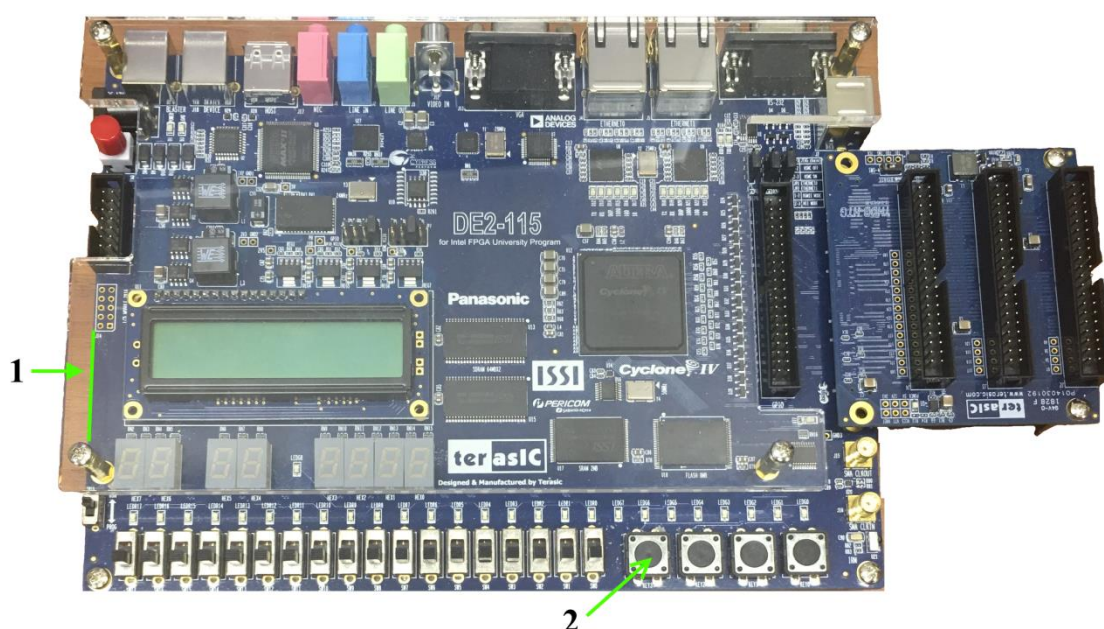


Рисунок 9. Подключение SD карты к тестеру

Так как слот для SD карт смонтирован с обратной стороны отладочной платы DE2-115, SD карту необходимо вставлять лицевой стороной вниз.

После того как SD карта вставлена в разъем, необходимо сбросить AVC1READER, чтобы AVC файл, записанный ранее, определился тестером. Для этого необходимо нажать на кнопку сброса RESET (Рисунок 9, 2).

Возможны следующие ситуации после нажатия на кнопку RESET со вставленной SD картой, которые отображаются на LCD дисплее (Рисунок 1, 3):

- “No SD Card! Press RESET” – SD карта не определяется по каким-либо причинам. Если после сброса это сообщение не пропадает, то это можно быть отсутствие контакта SD карты с тестером, ошибка контроллера SD карты, либо SD карта просто не поддерживается AVC1READER’ом;
- “No AVC File! Press RESET” – SD карта определилась, но файл AVC не был найден на ней. Возможно, по какой-то причине, файл не имеет расширения *.avc. Проверьте наличие файла с расширением *.avc, вставьте SD карту и заново нажмите RESET;
- “AVC:<имя AVC файла (первые 6 символов) >→1.AVC Press START” – SD карта определилась, и вместе с ней определился AVC файл, имя которого высвечивается на дисплее;

Когда SD карта и AVC файл полностью определились, можно начинать тестирование исследуемой микросхемы.

1.5 Запуск тестирования и поиск сигналов

Чтобы запустить тестирование микросхемы, необходимо убедиться, что шлейф подключен верно и все контакты идут к необходимым выводам микросхемы, прописанным в AVC файле.

Если все подключено правильно, то можно начинать тестирование, нажав на кнопку START (Рисунок 10, 1).

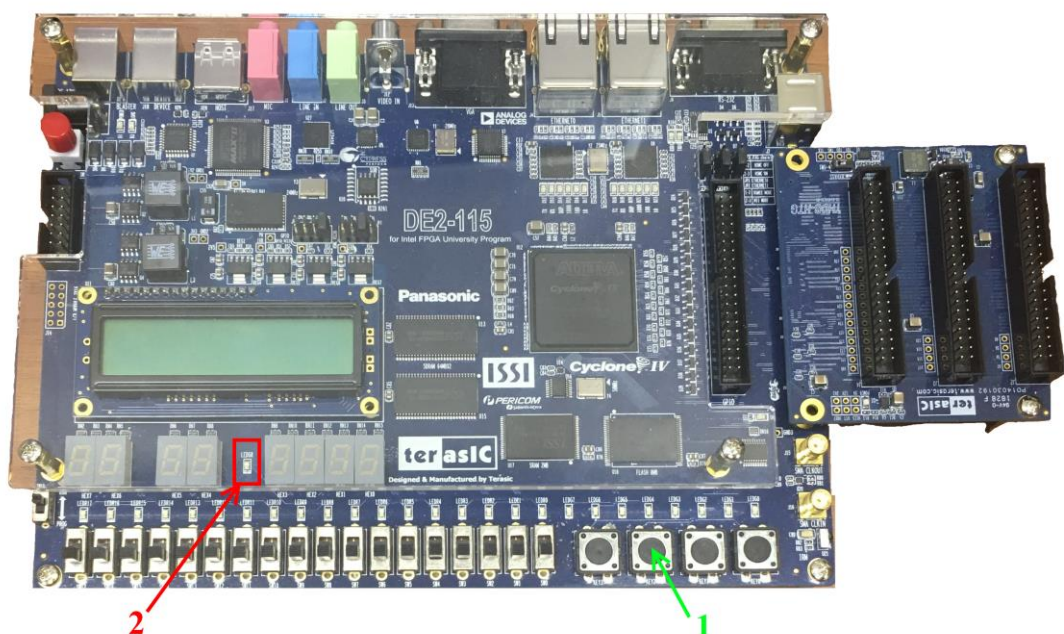


Рисунок 10. Запуск тестирования и индикация

Тестирование начинается с проверки AVC файла на двунаправленные и входные сигналы (поиск двунаправленных и входных сигналов). На LCD дисплее (Рисунок 1, 3) в это время отображается сообщение “*Searching for 3-state signals*”, а на 7-сегментных индикаторах поочередно загораются сегменты один за другим, отображая анимацию загрузки.

Процедура поиска может длиться долго, в зависимости от размера AVC файла. В среднем, поиск в файле размером 50 Мб может длиться около 30 секунд, если двунаправленный или входной сигнал находится в самом конце файла. С увеличением размера AVC файла, время поиска возрастает пропорционально, поскольку используется простой алгоритм прямого перебора. Время поиска можно снизить, если использовать все 4 линии шины DATA SD карты. Но в данной версии AVC1READER используется только 1 линия.

В AVC файле двунаправленные сигналы обозначены символами “H”, “L”, “X” и “Z”. То есть сигналы, которые могут быть входными либо переводятся во входное для тестера AVC1READER состояние. Только эти сигналы, которые приходят от исследуемой микросхемы на колодку IDC-40, сравниваются со значениями в AVC файле в данный момент времени. Если возникает ошибка, то они записываются в ERCY файл в виде векторов в соответствующем формате.

Если двунаправленных сигналов не было найдено в AVC файле, то тестеру AVC1READER нечего тестировать и, после поиска сигналов, на дисплее появляется сообщение “*No 3-state signal Press RESET*”. В это же время на всех 8-ми 7-сегментных индикаторах отображаются нули, а светодиод готовности LEDG8 (Рисунок 10, 2) мигает зеленым светом.

Чтобы поиск двунаправленных сигналов прошел удачно, убедитесь, что они присутствуют в AVC файле среди двоичных сигналов в виде символов, приведенных ранее. Вставьте обратно SD карту с записанным AVC файлом с двунаправленными сигналами, нажмите RESET, затем START.

Если поиск прошел успешно, за ним сразу же следует тестирование, заключающееся в чтении AVC, сравнении сигналов с микросхемы и в AVC файле и записи результатов в ERCY файл.

1.6 Процесс тестирования

Процесс тестирования микросхемы с помощью тестера AVC1READER представляет собой, как было сказано ранее, чтение векторов из AVC файла, содержащих двунаправленные сигналы, сравнение значений этих сигналов с сигналами поступающими с микросхемы и, при возникновении различий, запись ошибки в ERCY файл.

В процессе чтения векторов из AVC файла, происходит сравнение двунаправленных, если они переключены на вход, либо входных сигналов, с

сигналами следующими из тестируемой микросхемы. В это время на LCD дисплее (Рисунок 1, 3) отображается сообщение *“Reading AVC file signals...”*.

В случае возникновения различий при сравнении сигналов от микросхемы и сигнала в AVC файле, происходит запись каждой ошибки сначала в буфер, а затем запись всех ошибок в буфере в ERCY файл.

Во время записи ошибок в файл, на LCD дисплее отображается сообщение *“Writing ERCY vectors...”*, а на 7-сегментных индикаторах (Рисунок 1, 4) отображается количество ошибок, возникших с самого начала тестирования.

После окончания тестирования микросхемы, на LCD дисплее отображается информация об окончании процесса и проверки результатов (найденных ошибок) в ERCY файле *“Done! Check ERCY Press START”*. На 7-сегментном индикаторе отображается все количество ошибок, которое было найдено за все время тестирования микросхемы. Количество ошибок равно количеству векторов (строк) в ERCY файле. Светодиод готовности LEDG8 (Рисунок 9, 2) мигает зеленым светом.

Тестирование может длиться очень долго, всё зависит от количества найденных ошибок, размера AVC файла и количества задействованных линий данных SD карты. Как было отмечено ранее, в данной версии AVC1READER используется 1 линия данных.

К примеру, тестирование AVC файла размером 50 Мб (1,5 млн векторов и 15 сигналов), в каждом векторе которого содержится по 2 ошибки, которые потом записываются в ERCY файл, занимает около 15 мин.

После завершения тестирования можно переходить к обработке полученных результатов в ERCY файле.

1.7 Получение результатов

Теперь, чтобы получить ERCY файл, необходимо сбросить тестер, нажав на кнопку RESET, достать SD карту из слота и вставить её в кардридер ПК или внешний кардридер. Если SD карта не определилась с внешним кардридером, его необходимо переподключить со вставленной в него SD картой.

Сформированный во время тестирования микросхемы ERCY файл находится на SD карте в одной директории с записанным ранее AVC файлом. Файл ERCY имеет расширение *.erc и имя в формате E000000*.erc, где * - произвольный ASCII символ, начиная с символа “1” (0x31). Каждый новый тест увеличивает значение символа на единицу и так до переполнения, потом символ перебирается с самого нуля. Однако больше 1-2 тестов с 1 AVC файлом не требуется, поэтому старый ERCY файл стирается после записи нового AVC файла, а потом появляется после теста, но с индексом “1” (0x31).

Полученный после тестирования ERCY файл по своей структуре совпадает со стандартным ERCY файлом, получаемым с помощью промышленного тестера, но имеются небольшие различия по форме записи числовых значений, таких как номера векторов и сигналов с шин – числа

дополняются нулями. Возможно это вносит некоторые неудобства при чтении, но такова особенность реализации данной версии AVC1READER.

2. Отладочная индикация

Данный раздел содержит некоторые пояснения относительно отладочной информации AVC1READER.

На плате DE2-115 имеется набор из 26 светодиодов LEDG0 – LEDR17 (Рисунок 11, 1), который представляют собой отладочную индикацию машины состояний AVC1READER.

Каждый светодиод является индикатором какого-либо состояния машины состояний AVC1READER. С помощью машины состояний легко проводить диагностику тестера в случае какой-либо неисправности.

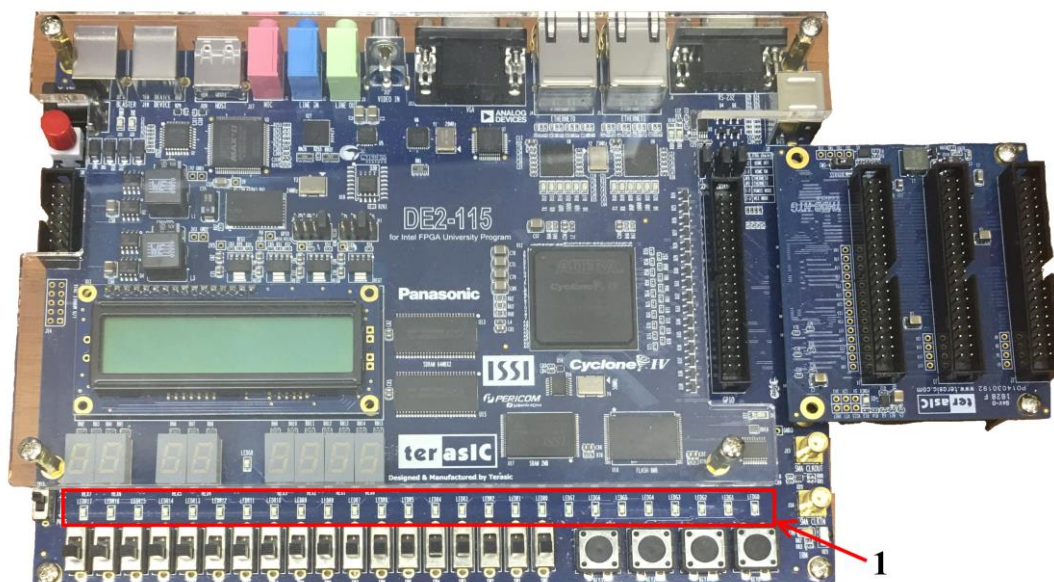


Рисунок 11. Отладочная индикация

Приведем по порядку значения состояний по каждому из светодиодов, названия (маркировка) которых нанесены на плате DE2-115:

- **LEDG0** – запись данных на SD карту в файл ERCY из предварительно записанного и заполненного буфера в ПЛИС;
- **LEDG1** – запись нулей на SD карту в конец файла ERCY;
- **LEDG2** – подтверждение записи данных на SD карту в ERCY файл;
- **LEDG3** – установка во внутреннем буфере ПЛИС параметров для файловой таблицы FAT32 для записи в неё информации о ERCY файле;
- **LEDG4** – запись файловой таблицы FAT32 из внутреннего буфера ПЛИС на SD карту вместо существующей таблицы;

- **LEDG5** – окончание всего процесса тестирования;
- **LEDG6** – установка параметров для чтения файловой таблицы FAT32 с SD карты с находящейся на ней записью о ERCY файле;
- **LEDG7** – чтение файловой таблицы FAT32 с SD карты с находящейся на ней записью о ERCY файле и сохранение таблицы во внутреннем буфере ПЛИС;
- **LEDR0** – команда CMD0 при инициализации SD карты;
- **LEDR1** – команда CMD8 при инициализации SD карты;
- **LEDR2** – команда ACMD41 при инициализации SD карты;
- **LEDR3** – команда CMD55 при инициализации SD карты;
- **LEDR4** – команда ответа R7 при инициализации от SD карты;
- **LEDR5** – команда ответа R1 при инициализации от SD карты;
- **LEDR6** – команда начала инициализации SD карты после подачи питания и сброса;
- **LEDR7** – команда CMD7 при инициализации SD карты и обращения к ней по адресу. Если это состояние не будет пройдено, то возникла ошибка контрольной суммы CRC7, так как это первая команда для которой формируется контрольная сумма;
- **LEDR8** – данный светодиод будет гореть постоянно, если произошла ошибка при начальной инициализации SD карты командой ACMD41. Это означает, что либо SD карта занята, либо она не поддерживается AVC1READER;
- **LEDR9** – запись FAT таблиц (FAT1 и FAT2) на SD карту из буфера в ПЛИС на место текущих FAT таблиц на SD карте;
- **LEDR10** – подтверждение записи FAT таблиц на SD карту из буфера в ПЛИС на место текущих FAT таблиц на SD карте;
- **LEDR11** – корректирование ранее прочитанных с SD карты и записанных буфер в ПЛИС FAT таблиц с учетом ранее записанного на SD карту ERCY файла;
- **LEDR12** – расчет конца новых FAT таблиц на SD карте с учетом ранее записанного ERCY файла;
- **LEDR13** – увеличение адреса чтения данных из AVC файла на SD карте;
- **LEDR14** – чтение данных из AVC файла на SD карте (по 512 байт);
- **LEDR15** – увеличение адреса чтения в FAT таблице при поиске записи об AVC файле;
- **LEDR16** – чтение найденной записи об AVC файле в FAT таблице;
- **LEDR17** – увеличение адреса при записи ERCY файла на SD карту;

Значения светодиодов могут изменяться от версии к версии AVC1READER. Данная версия предполагает вышеприведенные значения светодиодов.

Заключение

Данное руководство пользователя содержит полную информацию о принципах работы и использовании тестера микросхем на ПЛИС AVC1READER для обычного пользователя.

Текущая версия (ver. 1) не лишена недостатков с которыми, к сожалению, приходится пока мириться. К примеру, медленное чтение и запись, поскольку используется только 1 линия данных DATA SD карты. Также доставляет некоторое неудобство невозможность корректировать AVC файл прямо на SD карте, корректировать приходится только отдельно на машине с последующей перезаписью SD карты.

Также, возможно, недостатком является использование файловой системы FAT32, которая ограничивает использование файлов размером свыше 4 Гбайт. Однако, если AVC файлы не будут больше этого размера, то недостатком данный факт можно не считать.

Также была приведена отладочная информация, но для пользователя, возможно, она будет полезна при сообщении о каких либо неполадках разработчикам во время использования.