

## ПРИМЕНЕНИЕ ТЕХНОЛОГИИ SHMOO-ТЕСТИРОВАНИЯ В СИСТЕМЕ МОНИТОРИНГА ЭЛЕКТРОННЫХ УСТРОЙСТВ

**М.В. Водяницкий**

бакалавр

**О.Б. Богданова**

старший преподаватель

*Владивостокский государственный университет экономики и сервиса  
Владивосток, Россия*

*Данная статья описывает проблемы, возникающие при shmoo-тестировании оперативной памяти, и рассматривает возможность реализации системы мониторинга, для решения этих проблем.*

*Ключевые слова и словосочетания: shmoo-тестирование, оперативная память, система мониторинга.*

## APPLICATION OF SMOO TESTING TECHNOLOGY IN THE SYSTEM OF MONITORING OF ELECTRONIC DEVICES

*This article describes the problems that arise during shmoos testing of RAM, and considers the possibility of implementing a monitoring system to solve these problems.*

*Keywords: shmoos testing, random access memory, monitoring system.*

### **Введение**

Сегодня RAM (Random Access Memory), т.е. оперативная память используется во всех современных электронных вычислительных устройствах. Частоты работы потребительских устройств растут с каждым годом, техпроцесс все меньше и меньше, а сложность создания драйверов для таких устройств все выше и выше. Чем выше частоты и скорость работы, тем выше сложность правильной настройки для избежание нестабильной работы. Для портативных устройств все еще больше усложняется, ведь им требуется, как правило, работать в больших температурных диапазонах. Изменение температуры влияет на электрические характеристики полупроводников и может приводить к нестабильной устройства в целом.

В данной статье будет рассмотрен один из этапов процесса разработки драйверов для оперативной памяти – shmoos-тестирование, а также возможность применения этого тестирования в системе мониторинга.

### **Проблемы настройки оперативной памяти**

Оперативная память – одно из самых уязвимых мест, когда речь идет о быстро меняющейся температуре или влажности. Однажды настроенное устройство для работы при комнатной температуре, скорее всего не будет работать без дополнительного тестирования и настроек при минусовых температурах или в жару. Причина кроется в изменении электрических характеристиках чипа памяти и дорожек на устройстве, таких, как ЭДС, импеданс, сопротивление и прочих.

Помимо этого, из-за несовершенства изготовления во время фотолитографии, каждый чип получается немного индивидуальным. Большинство полупроводниковых изделий спроектированы с учетом того, что на кристалле могут возникать дефекты. Из-за этого возможности чипов могут различаться: какие-то экземпляры с одним и тем же партнамбером (идентификатор изделия) могут работать на нужной заявленной частоте при, допустим, 50 градусах Цельсия, а другие точно такие же будут работать нестабильно. Как правило, любая нестабильность работы памяти приводит к полной неработоспособности устройства в целом.

В итоге можно выделить несколько основных проблем:

- Температура влияет на настройку оперативной памяти;
- Индивидуальные особенности каждого отдельного чипа так же влияют на настройку.

Именно поэтому каждое новое устройство требует калибровки и подбора новых параметров. Подбор параметров DRAM происходит вручную и требует хорошего изучения спецификаций от производителя, проведения множества тестов с устройством. Однако возникает проблема с тем фактом, что есть определенный разброс характеристик разных чипов памяти и устройств в целом, к тому же зависящих от температурных условий. Настраивать вручную для каждого устройства память – задача не тривиальная. К тому же параметров слишком много даже для автоматизации этого процесса.

Описанные проблемы можно решить при помощи shmoо-тестирования.

### Shmoо-тестирование оперативной памяти

Первым делом для shmoо-тестирования подбирается наиболее чувствительный параметр к индивидуальным особенностям чипа при разных температурных условиях. Чаще всего таким параметром выступает DLL (Delay-locked Loop) [1]. Сам по себе DLL отвечает за контроль разницы между тактовым сигналом и сигналом входных данных. Настройка параметра DLL позволяет контролировать работу фазового детектора, который регулирует работу линии переменной задержки, используемой для синхронизации [2].

Основная задача shmoо-тестирования: подобрать один универсальный набор параметров DLL для всех устройств, позволяющий работать этим устройствам при требуемом (по спецификациям) диапазоне температур.

В результате shmoо-тестирования получается таблица: shmoо plot. Изначально shmoо plot подразумевает создание графиков тестирования электронных систем [3]. Пример на рисунке 1. На оси абсцисс отмечен тестируемый параметр, на оси ординат – тестируемое устройство. В качестве значения используется результат теста (например, нагрузочного), т.е. работало стабильно устройство, или нет.

	25	24	23	22	21	20	0	1	2	3	4	
у1	X	X	O	O	O	O	O	O	O	X	X	← значения
у2	X	X	X	O	O	O	O	O	X	X	X	
у3	X	X	X	X	O	O	O	O	X	X	X	
у4	X	X	X	O	O	O	O	X	X	X	X	
у5	X	X	X	X	O	O	O	O	O	O	X	
у6	X	X	O	O	O	O	O	O	O	X	X	

устройства

O	- пройденный параметр
X	- проблемный параметр
O	- оптимальный параметр

Рис. 1. Схема работы системы мониторинга

На таких графиках легко представить рабочий диапазон тестируемого параметра сразу для всех устройств.

Shmoо тестирование DRAM предполагает, что в качестве тестируемого параметра мы будем использовать набор значений DLL. Такими параметрами являются:

- Sync – настройка фазы синхронизации;
- Read – настройка фазы чтения (приема);
- Write – настройка фазы записи (отправки).

Все перечисленные параметры должны тестироваться последовательно и отдельно.

Для каждого устройства требуется последовательно менять значение параметра, проводить нагрузочный тест памяти (для выявления нестабильной работы), записывать результат нагрузочного теста. Весь shmo0-тест реализуется в виде специальной утилиты, которая запускается сразу после загрузки операционной системы тестируемого устройства. На рисунке 2 представлена блок-схема работы утилиты shmo0-тестирования.

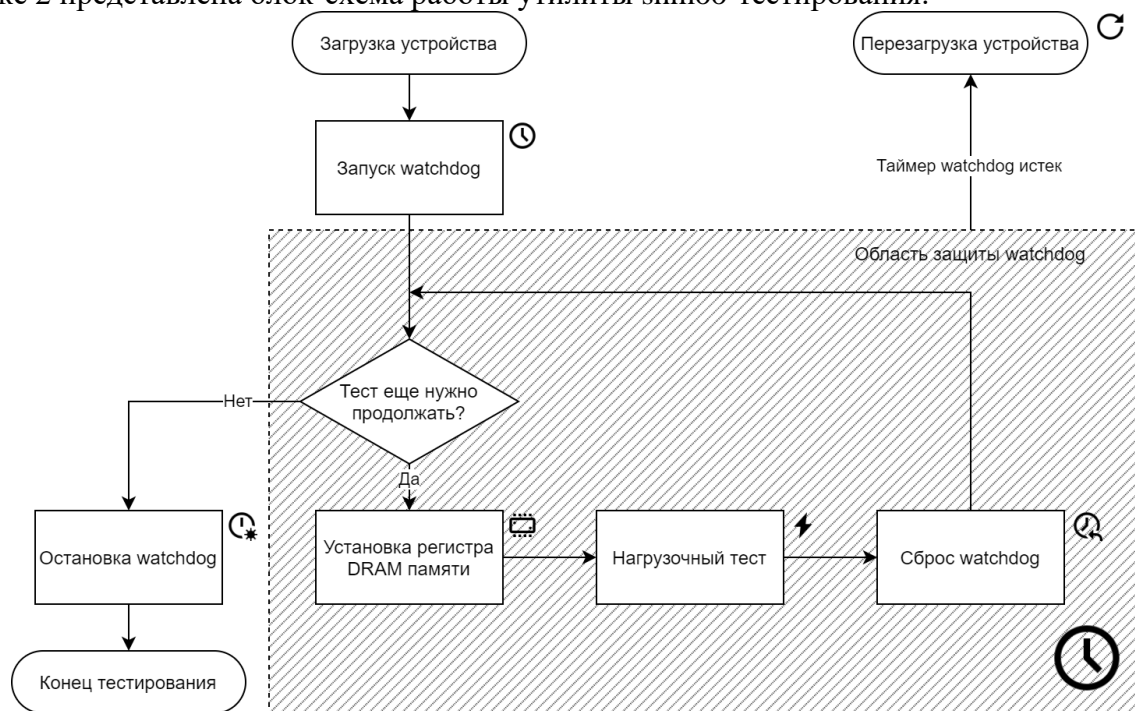


Рис. 2. Блок-схема работы утилиты shmo0-тестирования

После окончания такого теста получится набор протестированных параметров, который можно будет преобразовать в shmo0 plot.

### Проблемы реализации shmo0-тестирования в виде утилиты

В текущей реализации shmo0-тестирования, утилита запускается на тестируемом устройстве. Это приводит к ряду проблем:

- Не на всех устройствах присутствует возможность самостоятельно (т.е. программным способом) себя перезагрузить;
- Утилита работает на тестируемом устройстве и использует его DRAM, параметры которого сама же выставляет потенциально нестабильные, т.е. сама утилита может работать нестабильно.

Поэтому, если shmo0-тестирование реализовано в виде программной утилиты, инженеру, ответственному за проведения тестирования, придется постоянно и непрерывно наблюдать за процессом, так как может понадобится вручную перезагрузить зависшее устройство.

Помимо этого, сбор логов и контроль за всеми устройствами во время тестирования получается крайне сложным – ведь для мониторинга состояния тестирования приходится подключаться к каждому устройству, чтобы наблюдать за его логами.

Проанализировав данные проблемы, можно прийти к выводу, что процесс проведения shmo0-тестирования можно улучшить, путем создания отдельного устройства для мониторинга и проведения shmo0-тестирования.

### Система мониторинга shmo0-тестирования

В задачи системы мониторинга shmo0-тестирования будет входить: сбор логов по UART, обнаружение зависания устройства по тому же UART или через GPIO, перезагрузка устройства в случае зависания, удаленное информирование о состоянии проведения теста (сколько осталось до конца, сколько раз устройство уже перезагрузилось и т.д.). В таком устройстве должен быть предусмотрен удаленный доступ к логам и возможность ручного запуска, остановки или перезагрузки тестируемого устройства.

Система мониторинга должна быть независимой от тестируемого устройства, тем самым решить основные проблемы реализации shmoо-тестирования в виде утилиты. На рисунке 3 представлена схема системы мониторинга shmoо-тестирования.

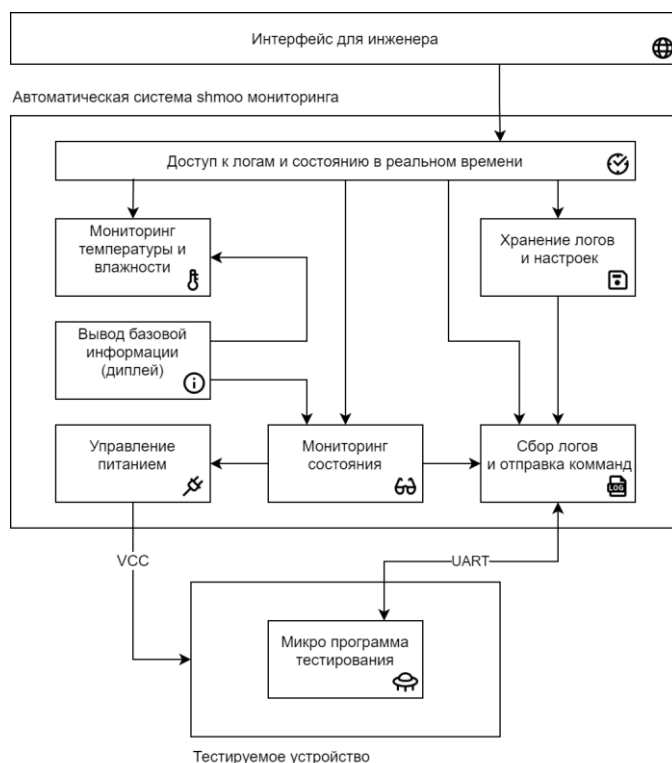


Рис. 3. Схема мониторинга shmoо-тестирования

Разработанная система мониторинга представлена на рисунке 4. В системе мониторинга используется микроконтроллер с возможностью подключения нескольких UART интерфейсов, GPIO для подключения сенсоров, I2C для дисплея и модуля SD карты.

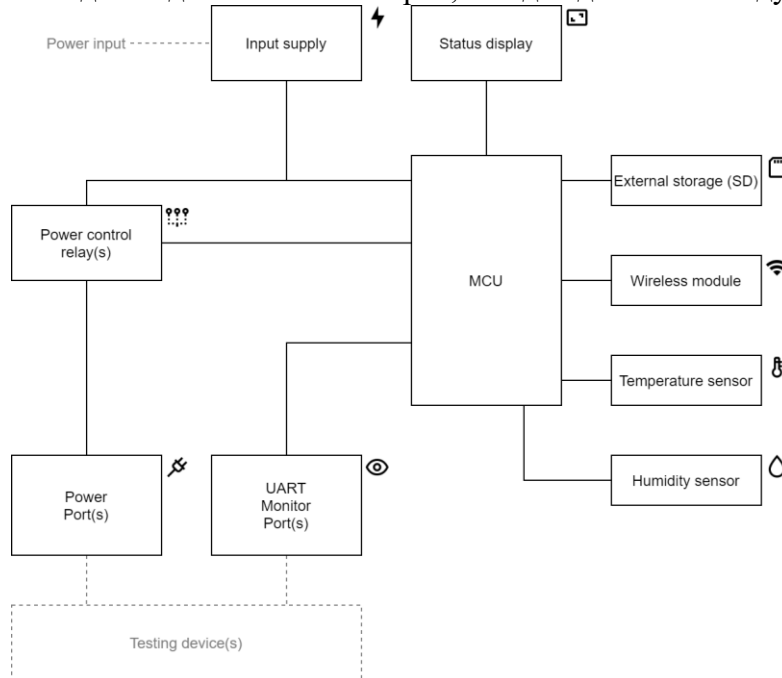


Рис. 4. Схема мониторинга shmoо-тестирования

Для управления питанием используется реле [4], управляя которым, микроконтроллер сможет включать, выключать и перезагружать тестируемые устройства. Система мониторинга должна быть оборудована беспроводным интерфейсом для реализации удаленного

интерфейса доступа к логам. Для хранения логов будет использоваться слот SD карты. Так же, для дополнительного вывода информации используется дисплей: на тот случай, если у инженера нет возможности подключиться к устройству.

Такое устройство будет проводить shmoо вместо утилиты на самом тестируемом устройстве.

### **Заключение**

Реализация shmoо-тестирования в виде отдельного независимого устройства системы мониторинга, позволит избежать множество трудностей, возникающих с классической реализацией подобного тестирования. Такое устройство позволит переложить большую часть работы с утилиты на внешнюю систему мониторинга, а также позволит добавить новые возможности, упрощающие и ускоряющие процесс разработки DRAM драйверов.

- 
1. Описание работы DLL и PLL [Электронный ресурс] – Режим доступа: <https://www.rambus.com/dllpll-on-a-dram/>
  2. Контур фазовой автоподстройки частоты и его основные свойства [Электронный ресурс] – Режим доступа: <http://www.dsplib.ru/content/pll/pll.html>
  3. Презентация на тему Shmoо [Электронный ресурс] – Режим доступа: [https://web.stanford.edu/class/archive/ee/ee371/ee371.1066/lectures/Old/lect\\_14\\_2up.pdf](https://web.stanford.edu/class/archive/ee/ee371/ee371.1066/lectures/Old/lect_14_2up.pdf)
  4. Описание видов реле и работа с ними – Режим доступа: <https://electrosam.ru/glavnaja/jelektrooborudovanie/rozetki-vykljuchатели/vidy-rele-i-primenenie/>