

# Принципы проектирования устройств тестового диагностирования быстродействующих микросхем и модулей полупроводниковой памяти

А.П. Евдокимов, В.Г. Рябцев, А.В. Меликов

Волгоградский государственный аграрный университет, akim.onoke@mail.ru

**Аннотация** — Рассматривается проблема повышения быстродействия устройств тестового диагностирования микросхем и модулей памяти. Предложена структура мультипроцессорного устройства тестового диагностирования, обладающего высоким быстродействием и обеспечивающего одновременное формирование тестовых воздействий для нескольких смежных тактов за один период сигнала синхронизации.

**Ключевые слова** — микросхема памяти, тестовое диагностирование, устройство тестового диагностирования.

## I. ВВЕДЕНИЕ

Вместе с микропроцессорами, быстродействие которых ежегодно возрастает, улучшаются также параметры полупроводниковых запоминающих устройств, что повышает общую эффективность вычислительной системы. Современные микросхемы памяти содержат программируемый контроллер, логический интерфейс приема-передачи данных, это позволяет отнести их к цифровым системам, содержащим встроенную память.

Для повышения технических характеристик микросхем памяти многие фирмы применяют новые технологии [1]. Например, фирма Samsung Semiconductor, Inc с 1998 года ведет разработку технологии полупроводниковых запоминающих устройств DDR SDRAM (Double Data Rate SDRAM), которая является логическим продолжением синхронной памяти (SDRAM). Микросхемы памяти DDR SDRAM были одобрены в качестве стандарта Объединенным техническим советом по электронным приборам (JEDEC) и в настоящее время изготавливаются многими фирмами [2]. Технология DDR позволяет записывать и считывать данные с частотой в два раза большей, чем частота системной шины, за счет передачи данных по обоим фронтам синхросигнала.

Частота передачи данных в микросхемах DDR1 составляет 400 МГц, в микросхемах DDR2 увеличена до 800 МГц, а в микросхемах DDR3 достигает 1800 МГц. В микросхемах памяти типа DDR4 частота достигает 3,2 ГГц. В 2018 году планируется выпуск микросхем типа DDR5, которые будут работать на частоте 3,2 ГГц, в недалеком будущем частота может увеличиться до 6,4 ГГц.

Эффективное тестовое диагностирование быстродействующих микросхем и модулей памяти можно выполнять только на реальной рабочей частоте. Однако максимальная частота устройств тестового диагностирования больших интегральных схем оперативных запоминающих устройств (БИС ОЗУ) ограничена быстродействием блока микропрограммного управления, который формирует микрооперации для алгоритмической генерации тестовых воздействий и эталонных реакций [3, 4].

Вместе с выпуском микросхем и модулей памяти нового типа проектируются тестеры для выполнения тестового диагностирования на рабочей частоте. Так частная компания CST Inc., занимающаяся проектированием тестеров для полупроводниковой памяти, анонсировала свой новый тестер Eureka DDR4 2400. Данный тестер способен тестировать модули DDR4 UDIMM / RDIMM / LRDIMM и SODIMM с широким диапазоном частот от 1600 до 2400 МГц. Тестер Eureka DDR4 2400 также может быть сопряжен с загрузчиком RoboFlex CST для автоматизации массового тестирования [5]. Однако структура и архитектура данного тестера и других аналогичных устройств не раскрыта в литературе подробно, что не позволяет производить их детальный анализ.

Отличительной особенностью тестеров для диагностирования микросхем и модулей памяти является алгоритмический способ формирования тестов под управлением блока микропрограммного управления (БМУ). Длительность минимального цикла БМУ определяется промежутком времени от момента выделения условий переходов до момента получения управляющих сигналов и вычисляется по формуле:

$$T_u = T_y + T_a + T_m + T_p + T_d;$$

где  $T_y$  – время выделения условий переходов, формируемых операционными секциями;

$T_a$  – время формирования адреса следующей микрокоманды;

$T_m$  – время выборки памяти микрокоманд;

$T_p$  – время переключения регистра микрокоманды;

$T_d$  – время дешифрации полей микрокоманды.

Целью настоящей работы является разработка принципов построения структуры и архитектуры

устройств тестового диагностирования быстродействующих микросхем и модулей полупроводниковой памяти.

## II. МЕТОД РАСПАРАЛЛЕЛИВАНИЯ ОПЕРАЦИИ ТЕСТОВОГО ДИАГНОСТИРОВАНИЯ ПОЛУПРОВОДНИКОВОЙ ПАМЯТИ

Значительного повышения быстродействия тестового диагностирования БИС ОЗУ можно достичь за счет распараллеливания процесса формирования векторов воздействий, последовательной передачи их при помощи коммутаторов на входы диагностируемой микросхемы памяти и параллельной обработки считанных реакций [6-8]. Для реализации данного метода повышения быстродействия предлагается в структуру устройства диагностирования микросхем и модулей памяти ввести группы блоков формирования адреса, блоков формирования данных и блоков сравнения. Учитывая, что число ячеек микросхем памяти кратно  $2^n$ , где  $n$  – любое целое положительное, поэтому количество блоков  $p = 2^r$ ,  $r = 0, 1, 2, 3, 4$  и т.д., что позволяет равномерно распределить ячейки между формирователями программ тестов.

Предлагаемый метод позволяет снизить требования по быстродействию к основным блокам устройства и увеличить частоту формирования тестовых воздействий и обработки эталонных реакций. При этом частота работы БМУ вычисляется по формуле:

$$f_{mпу} = f_d / p,$$

где  $f_{mпу}$  – частота работы блока микропрограммного управления;

$f_d$  – частота диагностирования микросхем памяти;

$p$  – коэффициент распараллеливания операций формирования векторов воздействий и обработки считанных реакций.

## III. АРХИТЕКТУРА МУЛЬТИПРОЦЕССОРНОГО УСТРОЙСТВА ТЕСТОВОГО ДИАГНОСТИРОВАНИЯ ПОЛУПРОВОДНИКОВОЙ ПАМЯТИ

Структура мультипроцессорного устройства тестового диагностирования (УТД) состоит из взаимодействующих управляющего и нескольких операционных процессоров:

$$M = \{УП, ОП_0, ОП_1, \dots, ОП_{p-1}\},$$

где УП – управляющий процессор;

$ОП_0, ОП_1, \dots, ОП_{p-1}$  – операционные процессоры;

$p$  – число параллельных операционных процессоров.

Каждый операционный процессор формирует код адреса диагностируемой запоминающей ячейки  $A_i$  и коды данных  $T_i$ , которые используются для задания входных данных объекта диагностирования при выполнении операции записи, а также как эталонные данные при выполнении операций сравнения ответных реакций. Структурная схема мультипроцессорного УТД, состоящего из управляющего процессора (УП) и нескольких операционных процессоров  $ОП_i = \{FT_i, FA_i\}$ , собранных из формирователей данных  $FT_i$  и формирователей адреса  $FA_i$ , приведена на рис. 1. Управляющий процессор формирует коды микроопераций, задающих режимы работы, для операционных процессоров и обеспечивает формирование вектора рабочих микроопераций:

$$C = \{W, R, XOR, M\},$$

где  $W$  – код микрооперации записи данных в диагностируемую ячейку;  $R$  – код операции считывания данных из диагностируемой ячейки;  $XOR$  – код операции сравнения считанных и эталонных данных;  $M$  – код маски, обычно используется для управления электронными узлами согласования устройства и объекта диагностирования.

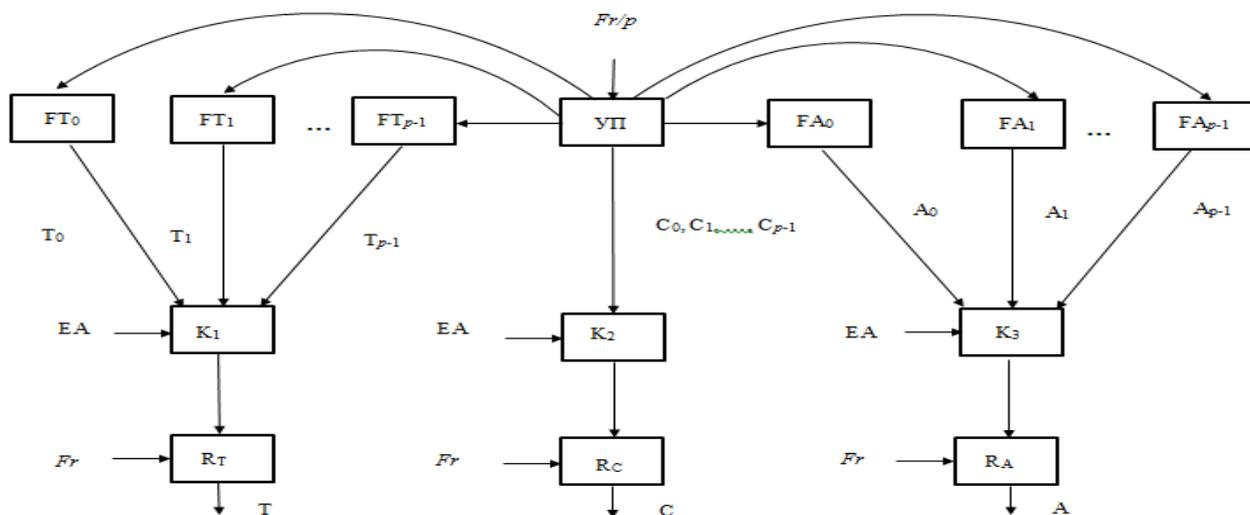


Рис. 1. Структурная схема устройства тестового диагностирования



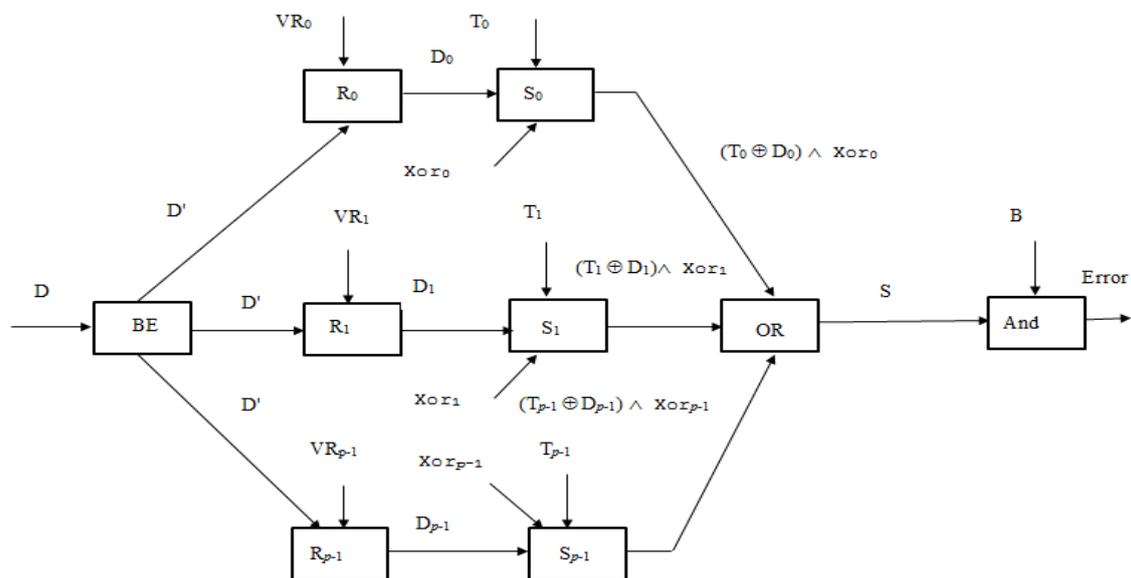


Рис. 2. Структура блока параллельной обработки считанных реакций

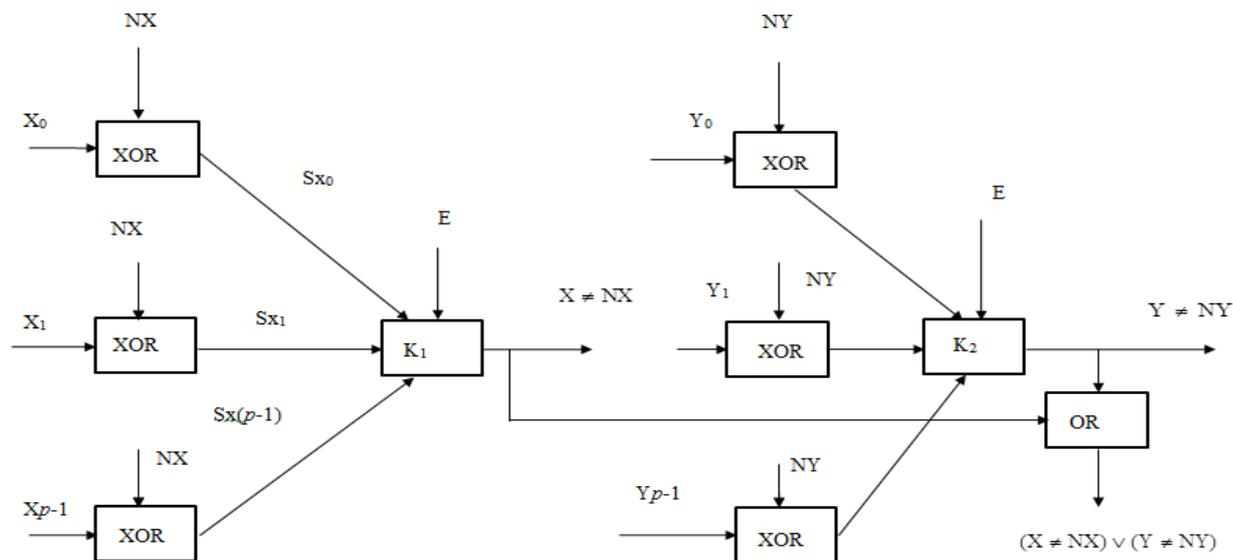


Рис. 3. Структура блока формирования условий переходов, учитывающих коды NX и NY

Выбор условных переходов обеспечивают коммутаторы K1 и K2, на адресные входы которых подается двоичный код E, соответствующий номеру адресного формирователя, сигналы схем сравнения которого используется в данном такте диагностирования.

Для реализации команд условных переходов, обеспечивающих обнаружение соответствия кодов текущих адресов X и Y кодам конечных адресов NX, NY, или начальных адресов GX или GY, признаки переходов вычисляются при помощи выражений:

$$F_{nx} = \begin{cases} (X_0 \neq NX) \Leftarrow E=0; \\ (X_1 \neq NX) \Leftarrow E=1; \\ \dots \\ (X_{p-1} \neq NX) \Leftarrow E=p-1; \end{cases}$$

$$F_{ny} = \begin{cases} (Y_0 \neq NY) \Leftarrow E=0; \\ (Y_1 \neq NY) \Leftarrow E=1; \\ \dots \\ (Y_{p-1} \neq NY) \Leftarrow E=p-1; \end{cases}$$

$$F_{gx} = \begin{cases} (X_0 \neq GX) \Leftarrow E=0; \\ (X_1 \neq GX) \Leftarrow E=1; \\ \dots \\ (X_{p-1} \neq GX) \Leftarrow E=p-1; \end{cases}$$

$$F_{gy} = \begin{cases} (Y_0 \neq GY) \Leftarrow E=0; \\ (Y_1 \neq GY) \Leftarrow E=1; \\ \dots \\ (Y_{p-1} \neq GY) \Leftarrow E=p-1. \end{cases}$$

Аналогичные выражения можно получить для вычисления значения флагов, учитывающих коды NX и NY или комбинации кодов  $(X \neq NX) \neq (Y \neq NY)$ ,  $(X \neq GX) \neq (Y \neq GY)$ , которые является дизъюнкцией двух рассмотренных выше условий. Формирователи кода адреса генерирует вектор адреса  $A = \{X, Y\}$ , который обеспечивает выборку строк X и столбцов Y ячеек памяти. Информационные данные T, которые включаются в вектор тестовых воздействий, используются также в качестве кодов эталонных реакций.

#### IV. ПРОГРАММНАЯ МОДЕЛЬ МИКРОПРОГРАММНЫХ СЧЕТЧИКОВ

При выполнении операции считывания данные из выбранной ячейки памяти заносятся в регистры реакций  $Dout_0, Dout_1, \dots, Dout_{p-1}$ . Разрешение записи данных обеспечивают синхронизирующие сигналы, которые разнесены по времени друг от друга на период такта обращения к диагностируемому изделию. Считанные и эталонные данные сравниваются схемами сравнения  $S_0, S_1, \dots, S_{p-1}$ , на выходах которых формируется результат диагностирования, который может маскироваться кодом специального регистра В.

Формирование тестовых воздействий обеспечивается алгоритмическим способом по командам УП, программная модель счетчиков которого приведена на рис. 4.

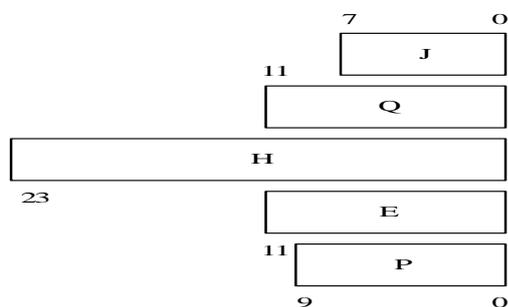


Рис. 4. Программная модель счетчиков управляющего процессора

Количество и число разрядов счетчиков определяется максимальным количеством циклов повторения микроопераций в наиболее распространенных тестах. Максимальное значение переменных, которые заносятся в счетчики J, Q, H, определяется по формулам:

$$J = \log_2 n - 3, Q = \sqrt{n} - 1, H = n/2 - 1, E = (\sqrt{n} - 4) / 2 - 1,$$

где  $n$  – емкость памяти диагностируемого изделия.

Количество разрядов счетчика адреса команд P определяется емкостью микропрограммной памяти, необходимой для хранения программ тестов.

#### V. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ ОСНОВНЫХ КОМПОНЕНТОВ УТД

В среде Active-HDL разработан проект УТД, структура которого приведена на рис. 5 в составе: микроконтроллер, счетчик адреса микрокоманд, блок формирования адреса, блок формирования данных, компаратор данных и блок формирования флагов условных переходов. На входы микроконтроллера, поступают коды флагов переходов  $Flag(2:0)$ , микрокоманды  $Com(3:0)$ , сигнал сброса RST и тактовые импульсы CLK. Микроконтроллер осуществляет дешифрацию кодов микрокоманды и формирует микрооперации, определяющие режимы работы остальных блоков УТД. Выполненная верификация проекта подтвердила его работоспособность.

Разработку программ диагностирования изделий полупроводниковой памяти для устройства, имеющего мультипроцессорную структуру, предлагается осуществлять на машиноориентированном языке программирования Prover, синтаксис которого приведен ниже.

#### СИНТАКСИС МАШИНООРИЕНТИРОВАННОГО ЯЗЫКА PROVER

Следует выделить основные микрооперации, которые при тестировании полупроводниковой памяти для повышения быстродействия надо выполнять параллельно:

- а) формирование кода адреса;
- б) формирование кода данных;
- в) рабочие микрооперации записи (W), считывания (R) и сравнения считанных и эталонных данных (A).

Для разработки программ тестового диагностирования запоминающих устройств создан язык программирования Prover, обеспечивающий задание необходимых команд и микроопераций, которые записываются в соответствии с синтаксисом, приведенном в табл. 1.

Таблица 1

Формат микроопераций языка Prover

| Позиции экрана при записи команд |   |   |   |                                    |   |     |    |    |  |  |
|----------------------------------|---|---|---|------------------------------------|---|-----|----|----|--|--|
| 1                                | 2 | 3 | 4 | 5                                  | 6 | ... | 29 | 30 |  |  |
| Метка, номер такта               |   |   |   | Микрооперации, разделенные запятой |   |     |    |    |  |  |

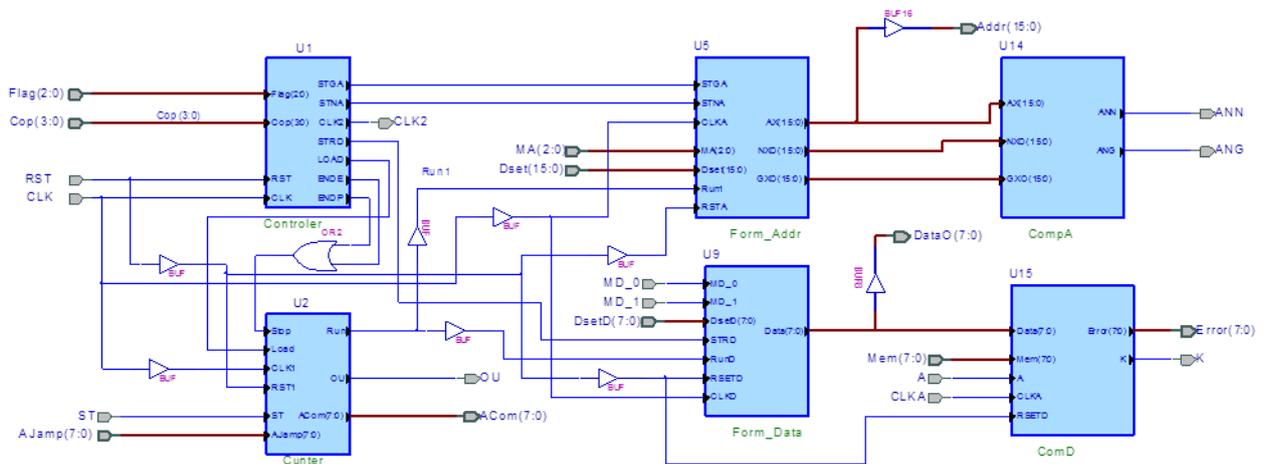


Рис. 5. Структура проекта УТД

В первых трех позициях допускается использовать метки команд, по которым осуществляются передачи управления или номер такта, в котором осуществляется выполнение микроопераций. В четвертой позиции всегда должен размещаться разделительный пробел. В позициях с 5 по 30 помещаются микрооперации, которые разделяются друг от друга запятыми. Символ пробела в этих позициях обозначает окончание команды.

Для формирования кодов адреса применена двумерная система координат (X, Y), что удобно для выбора строк и столбцов запоминающих ячеек. Для синтеза программ применен набор микроопераций формирования кода адреса, приведенный в табл. 2.

Таблица 2

Формат микроопераций изменения кода адреса

| Микро-операции | Выполняемые действия                                 |
|----------------|--|
| X:=GX          | Запись начального адреса GX в X                      |
| X:=X           | Сохранение содержимого X                             |
| X:=X+1         | Увеличение X на единицу                              |
| X:=X+2         | Увеличение X на 2                                    |
| X:=X+3         | Увеличение X на 3                                    |
| X:=X+4         | Увеличение X на 4                                    |
| X:=NX          | Запись конечного адреса NX в X                       |
| Y:=NY          | Запись конечного адреса NY в Y                       |
| Y:=GY          | Запись начального адреса GY в Y                      |
| Y:=Y           | Сохранение содержимого Y                             |
| Y:=Y+1         | Увеличение Y на 1                                    |
| Y:=Y+2         | Увеличение Y на 2                                    |
| Y:=Y+3         | Увеличение Y на 3                                    |
| Y:=Y+4         | Увеличение Y на 4                                    |
| Y:=Y+1*        | Увеличение Y на 1, если есть перенос из координаты X |
| Y:=Y-1*        | Уменьшение Y на 1, если есть заем из координаты X    |

Для формирования кодов данных реализованы следующие микрооперации:

- сохранение данных (T:=T);
- занесение исходных данных (T:=D);
- инверсия данных (T:=NOTT).

Для кодирования параллельно выполняемых микроопераций в команде выделены отдельные операционные поля. Число полей определяется количеством совместно выполняемых микроопераций. Для циклического повторения фрагментов тестов используются следующие признаки ветвления программы: X≠NX, Y≠NY, X≠GX, Y≠GY и комбинации данных условий: X≠NX и Y≠NY, X≠GX и Y≠GY.

Пример 1.

```

MET JXYN, MET, 1
0 W, X:=X+1, Y:=Y+1*
1 R, X:=X+2>X, Y:=Y+1*
3 A, X:=X+2, Y:=Y+1*

```

В данном примере микрооперации в тактах 0,1 и 3 будут выполняться до тех пор, пока выполняется условие X≠NX и Y≠NY. При невыполнении указанного выше условия микрооперации выполняются последний раз, и осуществляется переход к следующей команде.

Команды занесения данных устанавливают исходные состояния программных регистров, начальные и конечные адреса диагностируемой памяти коды регистров данных. Установка исходных значений регистров GX, GY определяет начальный адрес диагностируемого объема памяти. Запись данных в регистры NX, NY обеспечивает указание конечного адреса диагностируемого изделия.

Команды условных переходов выполняются совместно с микрооперациями сопутствующих им тактов. Сопутствующие такты определяются следующим образом: первый такт, следующий за командой условного перехода всегда сопутствующий;

последующий такт считается сопутствующим, если его номер больше предыдущего (всего 4 такта, т.е. максимальное количество сопутствующих тактов равно четырем).

Переходы в этих командах осуществляются при выполнении соответствующих условий. Для поиска условий в команде может быть указан такт, по которому проводится эта проверка. При отсутствии номера такта интерпретатор выдает сообщение об ошибке. Таким образом, в программе необходимо предусмотреть переменную, указывающую номер такта проверки. При невыполнении условий перехода ни в одном такте интерпретатор переходит к рассмотрению команды или микроопераций, следующих за данной командой условного перехода. Если же условия выполнены, программа производит следующие действия: запоминает метку, на которую необходимо осуществить переход, выполняет микрооперации в сопутствующих команде тактах и выполняет переход на требуемую метку.

При работе интерпретатора в памяти компьютера выделяется участок объемом  $(NX+1)*(NY+1)$ . В данный участок памяти может производиться запись различных кодов данных.

По алгоритму отлаживаемого теста данные считываются и сравниваются с эталонными значениями. При выполнении команды сравнения данных, считанных из выбранной ячейки, и эталонного значения счётчик К переключается в 1, если эти значения различны, т.е. при возникновении ошибки. Регистр ошибок фиксирует одиночные ошибки в разрядах данных, что позволяет по его значению в конце выполнения теста определить, было ли произведено хотя бы одно ошибочное считывание данных при диагностировании изделия.

Команды останова отображают конечный результат выполнения программы теста. Индикация при останове определяется по значению счетчика К – указателя ошибки: ENDP – останов с индикацией «ГОДЕН»; ENDE – останов с индикацией «БРАК».

#### Выводы

В предлагаемой структуре УТД частота формирования тестов определяется только временем переключения коммутаторов и выходных регистров и задержками времени распространения сигналов по линиям связи. Предложенный принцип построения УТД полупроводниковой памяти может быть применен для проектирования комплексов тестового диагностирования цифровых систем, содержащих встроенную память. Формат команд языка Prover

позволяет разрабатывать программы тестов для запоминающих устройств с различной структурой, что обеспечивает тестирование изделий полупроводниковой памяти с широким диапазоном основных параметров.

#### ПОДДЕРЖКА

Работа выполнена при финансовой поддержке гранта РФФИ (проект 16-08-00393).

#### ЛИТЕРАТУРА

- [1] Lee, Seung-Joo, Dynamic Capabilities at Samsung Electronics: Analysis of its Growth Strategy in Semiconductors (August 22, 2011). Available at SSRN: <https://ssrn.com/abstract=1914116>
- [2] Main Memory: DDR4 & DDR5 SDRAM. [Electronic resource]. Access mode: <https://www.jedec.org/category/technology-focus-area/main-memory-ddr3-ddr4-sdram>
- [3] Yarmolik S. V., Yarmolik V. N. Nondestructive RAM testing based on multiple signature comparison // Automatic Control and Computer Sciences, August 2009, Volume 43, Issue 4. – Pp. 173-178.
- [4] Muddapu Parvathi, N. Vasantha, Satya Parasad Modified March C - Algorithm for Embedded Memory Testing// International Journal of Electrical and Computer Engineering (IJECE) Vol. 2, № 5, October 2012. – Pp. 571-576.
- [5] CST Start to Deliver Eureka 2400 DDR4 Memory Tester/ [Electronic resource]. Access mode: <http://www.simmtester.com/page/news/showcstnews.asp?num=120>
- [6] Almadi M.K., Moamar D.N., Ryabtsev V.G. Methodology of Algorithms Synthesis of Memory Test Diagnosing // Proceedings of IEEE East-West Design & Test Symposium 2010 (EWDTS'10). St. Petersburg, 17-20 September 2010. – Pp. 366-370.
- [7] Almadi M.K., Ryabtsev V.G. New Infrastructure for Memory Tests Design // Proceedings of the International Workshop Critical Infrastructure Safety and Security (CrISS-DESSERT 2011). Kirovograd, May 11-13, 2011. – Pp. 434-440.
- [8] Ryabtsev V., Evseev K., Almadi M. The Concept of Memory Device Diagnosis Algorithm Design// Journal of Multidisciplinary Engineering Science and Technology (JMEST), Vol. 3, Issue 10, October. 2016. – Pp. 5771-5774.
- [9] Ryabtsev V.G., Kudlaenko V.M, Movchan Y.U. Method of estimation diagnostic properties of the Test Family March. // Proceeding of East-West & Test International Workshop (EWDTW'04).Yalta-Alushta, Criema, Ukraine. September 23-26, 2004. – Pp. 220-224.
- [10] Рябцев В.Г., Шубович А.А., Евсеев К.В. Автоматизированное проектирование дискретных тестов диагностирования запоминающих устройств / Современные наукоемкие технологии, № 6, 2016. – С. 288-294.

# Principles of Designing Devices for Test Diagnosing of High-speed Microchips and Semiconductor Memory

A.P. Evdokimov, V.G. Ryabtsev, A.V. Melikov

Volgograd Agrarian State University, akim.onoke@mail.ru

**Abstract** — The article describes applying parallelization of the formation process of forcing vectors to improve the performance of test devices for diagnosing microcircuits and memory modules. Their sequential transfer to the inputs diagnosed device is by switches. In this article, it is parallel processes of reading responses.

The proposed method makes it possible to increase the frequency of the test actions formation and the processing of reading responses  $p$ -times, where  $p$  – is the coefficient of operation parallelization.

The structure of the multiprocessor diagnostic test device consists of the interacting control and  $p$  operation processors. Each operational processor generates the address code of the diagnosed memory cell  $A_i$  and the data codes  $T_i$ , which are used to specify the input data of the diagnostic object when the write operation is performing. The operational processors contain data generation units  $FT_i$  and address generation units  $FA_i$  that generate test impacts for several next times of diagnostic.

Transfer flags are formed because of comparing the codes of the current addresses in  $X$  and  $Y$  coordinates with  $NX$ ,  $NY$  end address codes or  $GX$ ,  $GY$  begin address codes. It does for implementing conditional transfer commands that provide access to all cells of the tested memory.

When the read operation is performed, data from the selected memory cell is recorded in the registers of reaction  $Dout_0, Dout_1, \dots, Dout_{p-1}$ . Synchronizing signals provides resolution of data recording. They have spaced apart from each other for the period of the access time to the diagnosed product. The read and reference data are compared by comparison schemes  $S_0, S_1, \dots, S_{p-1}$ , at the outputs of which the diagnostic result is generated. It can be masked by the code of the particular register  $B$ .

The project of a test diagnosis device has been developed in the Active-HDL. It consists of a microcontroller, an address counter of micro-operations, an address generation unit, a data generation unit, a data comparator, and a conditional transfer flags generation unit. The completed verification of the project confirmed its operability.

In the proposed structure of test diagnosis device, the test generation frequency is determined only by the switching time and the output registers and depends on the propagation time of the signals through the communication line. The proposed principle of constructing a test diagnosis device can be applied for designing of complexes for test diagnostic of digital systems containing built-in memory.

**Keywords** — memory chip, test diagnostics, diagnostic test device

## PROJECT SUPPORT

The research is powered by RFBR grant financial support (project 16-08-00393).

## REFERENCES

- [1] Lee, Seung-Joo, Dynamic Capabilities at Samsung Electronics: Analysis of its Growth Strategy in Semiconductors (August 22, 2011). Available at SSRN: <https://ssrn.com/abstract=1914116>
- [2] Main Memory: DDR4 & DDR5 SDRAM. [Electronic resource]. Access mode: <https://www.jedec.org/category/technology-focus-area/main-memory-ddr3-ddr4-sdram>
- [3] Yarmolik S. V., Yarmolik V. N. Nondestructive RAM testing based on multiple signature comparison // Automatic Control and Computer Sciences, August 2009, Volume 43, Issue 4. – Pp. 173-178.
- [4] Muddapu Parvathi, N. Vasantha, Satya Parasad Modified March C - Algorithm for Embedded Memory Testing // International Journal of Electrical and Computer Engineering (IJECE) Vol. 2, № 5, October 2012. – Pp. 571-576.
- [5] CST Start to Deliver Eureka 2400 DDR4 Memory Tester/ [Electronic resource]. Access mode: <http://www.simmtester.com/page/news/showcstnews.asp?num=120> (December 28, 2017).
- [6] Almadi M.K., Moamar D.N., Ryabtsev V.G. Methodology of Algorithms Synthesis of Memory Test Diagnosing // Proceedings of IEEE East-West Design & Test Symposium 2010 (EWDTS'10). St. Petersburg, 17-20 September 2010. – Pp. 366-370.
- [7] Almadi M.K., Ryabtsev V.G. New Infrastructure for Memory Tests Design // Proceedings of the International Workshop Critical Infrastructure Safety and Security (CrISS-DESSERT 2011). Kirovograd, May 11-13, 2011. – Pp. 434-440.
- [8] Ryabtsev V., Evseev K., Almadi M. The Concept of Memory Device Diagnosis Algorithm Design // Journal of Multidisciplinary Engineering Science and Technology (JMEST), Vol. 3, Issue 10, October. 2016. – Pp. 5771-5774.
- [9] Ryabtsev V.G., Kudlaenko V.M., Movchan Y.U. Method of estimation diagnostic properties of the Test Family March. // Proceeding of East-West & Test International Workshop (EWDTW'04). Yalta-Alushta, Criema, Ukraine. September 23-26, 2004. – Pp. 220-224.
- [10] Rjabcev V.G., Shubovich A.A., Evseev K.V. Avtomatizirovannoe proektirovanie diskretnyh testov diagnostirovaniya zapominajushhiih ustrojstv / Sovremennye naukoemkie tehnologii, № 6, 2016. – C. 288-294.