

УДК 004.021

АЛГОРИТМИЧЕСКИЙ МЕТОД ВЕРИФИКАЦИИ VERILOG-МОДЕЛЕЙ МИКРОСХЕМ АСИНХРОННОЙ ПАМЯТИ

Моамар Д.Н., Рудь Ю.Г., Рябцев В.Г.

Черкасский государственный технологический университет
кафедра специализированных компьютерных систем

E-mail: Yulsee@rambler.ru

Аннотация

Моамар Д.Н., Рудь Ю.Г., Рябцев В.Г. Алгоритмический метод верификации Verilog-моделей микросхем асинхронной памяти. Предлагается алгоритмический способ проектирования тестов для верификации Verilog-моделей микросхем памяти, что обеспечивает сокращение трудоемкости формирования тестовых воздействий и эталонных реакций. Определен подход для автоматического фиксирования и сравнения данных.

Общая постановка проблемы

При проектировании управляющих и информационных систем важной задачей является выбор типа микросхем для построения оперативных запоминающих устройств. В настоящее время широкое распространение получили микросхемы асинхронной оперативной памяти, основными параметрами которых являются емкость и число разрядов данных [1]. Для реализации контроллера запоминающего устройства необходимо выполнить требования к последовательности формирования управляющих сигналов, которые указаны в технической документации. Наиболее целесообразно проверку корректности формирования данных сигналов выполнять при верификации Verilog-моделей микросхем памяти, которые свободно распространяются в сети Internet.

При выполнении моделирования микросхем асинхронной оперативной памяти в известных методах предлагается записывать коды данных в запоминающие ячейки, выполнять операции считывания и выводить на экран монитора или записывать в файлы состояния управляющих сигналов, коды адреса и коды записываемых и считываемых данных [2]. При этом последовательно для каждой ячейки в тесте указываются коды адреса и данных, что при большом объеме запоминающего устройства является довольно трудоемкой процедурой. Кроме того достоверность считанных данных необходимо подтверждать визуальным сравнением считанных и эталонных данных, что усложняет процесс верификации Verilog-моделей микросхем памяти и может привести к ошибкам при проектировании аппаратных средств, реализующих интерфейс запоминающего устройства.

Целью данной работы является разработка метода, обеспечивающего снижение трудоемкости синтеза программ тестов для верификации Verilog-моделей микросхем памяти, что сократит сроки проектирования управляющих и информационных систем.

Исследования

Для задания объема и числа разрядов микросхемы асинхронной оперативной памяти в Verilog-модели используются следующие параметры: `dqbits` – ширина шины данных; `addbits` – ширина шины адреса; `memdepth` – емкость памяти. Для связи с внешней средой микросхема памяти содержит следующие порты:

```
input CE_, OE_, WE_, LB_, UB_;
input [(addbits - 1) : 0] A;
inout tri [(dqbits - 1) : 0] IO;
```

Для подключения модели микросхемы к испытательному стенду предлагается применять схему подключения, приведенную ниже.

```
IS61LV6416L UUT (
    .A(A),
    .IO(IO_bidir),
    .CE_ (CE_ ),
    .OE_ (OE_ ),
    .WE_ (WE_ ),
    .LB_ (LB_ ),
    .UB_ (UB_ ));
```

При формировании теста при помощи встроенных средств стимулирования среды Active-HDL необходимо осуществлять останов процесса верификации модели для переключения сигналов шины ввода/вывода в третье (отключенное) состояние. Это действие необходимо выполнять при работе стимулятора в режиме Drive, когда допускается воздействие на текущие сигналы еще одного драйвера. Дальше можно продолжать верификацию модели в режиме считывания данных. Макрос, формируемый средой Active-HDL, создает тестовую последовательность, приведенную ниже и содержащую только сигналы адреса и данных без указания необходимых для работы микросхемы управляющих сигналов.

```
initial
begin : STIMUL // begin of stimulus process
    #0
    A = 3'b000;
    IO = 8'b00000000;
#100000; //0
    A = 3'b001;
    IO = 8'b00000001;
#100000; //100000
    A = 3'b010;
    IO = 8'b00000010;
#100000; //200000
    A = 3'b011;
    IO = 8'b00000011;
.....
end // end of stimulus process
```

Реализация аппаратных средств, реализующих связь микросхемы с процессором, при помощи такого теста затруднительна. Для задания модельного времени на языке Verilog применяется команда `timescale 1ps/1ps, где первый параметр задает шаг моделирования, а второй – точность установки модельного времени. Для записи кода данных, равного коду адреса в весь объем микросхемы, предлагается циклически выполнять следующую последовательность команд:

```
for (n =0 ; n <= memdepth; n = n+1)
begin
    CE_ = 0;
    OE_ = 1;
    LB_ = 0;
    UB_ = 0;
#10000 WE_ = 0;
#100000 WE_ = 1;
A=A+1;
IO=IO+1;
end
```

Временная диаграмма работы микросхемы в режиме записи данных приведена на рис. 1 и иллюстрирует изменение кодов адреса и данных, а также состояние управляющих сигналов, задающих режим работы микросхемы.

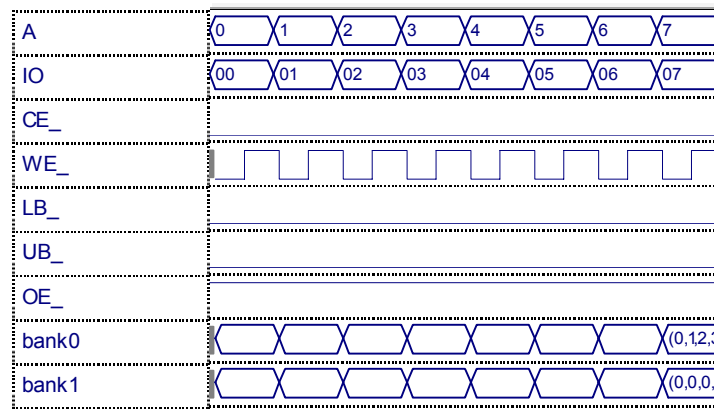


Рисунок 1 - Временная диаграмма работы микросхемы в режиме записи данных

Такой метод формирования входных воздействий по предлагаемому алгоритму сокращает объем программы и позволяет создать тест для микросхемы заданного объема путем изменения величин ее основных параметров [3, 4].

Для формирования эталонных и считанных данных предлагается использовать дополнительные регистры, описание которых приведено ниже.

```
reg [dqbits-1:0] IO_C;
reg [dqbits-1:0] Mem;
```

Для записи считанных данных в регистр Mem используется специальный сигнал Strobe, положение которого определяется временем выборки данных, заданным параметром Taa. По переднему фронту данного строба сигналы IO_bidir, считанные из ячеек микросхемы памяти, запоминаются в регистре Mem и используются затем для сравнения с эталонными данными IO_C. Временная диаграмма, иллюстрирующая работу микросхемы памяти во всех режимах, показана на рис. 2.

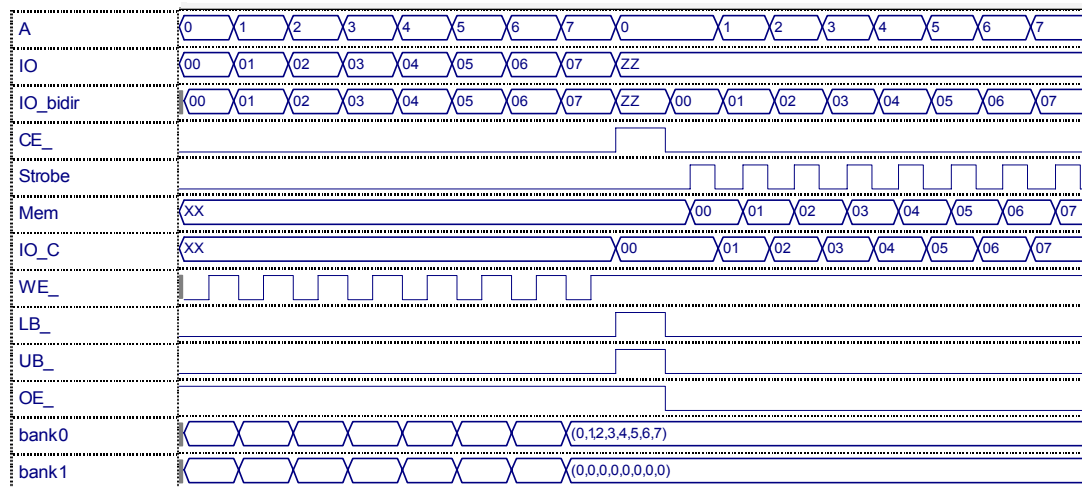


Рисунок 2 - Общая диаграмма работы микросхемы памяти

Фрагмент программы теста, команды которого обеспечивают работу микросхемы в режиме считывания и сравнения данных, приведен ниже.

```
begin
  for (n =0; n <= memdepth; n=n+1)
    begin
      CE_ =0;
      OE_ = 0;
      LB_ = 0;
      UB_ = 0;
      WE_ = 1;
      # (Taa+50000) Strobe = 1;
      if (Strobe == 1) Mem = IO_bidir;
      # 50000 Strobe = 0;
      if (Mem != IO_C)
        # 0 $display ("Результат Брак");
      #5000 WE_ = 1;
      A=A+1;
      IO_C=IO_C+1;
    end
  end
  # 0 $display ("Конец теста");
```

Если считанные и эталонные данные различаются между собой, то на экран монитора выдается сообщение "Результат Брак". При успешном завершении теста такое сообщение не выдается, а появляется запись "Конец теста", что свидетельствует о завершении верификации модели микросхемы памяти.

Выводы

За счет применения алгоритмического способа проектирования тестов сокращается трудоемкость формирования тестовых воздействий и эталонных реакций для верификации Verilog-моделей микросхем памяти асинхронного типа. Автоматическое фиксирование и сравнение считанных и эталонных данных исключает операцию визуального сравнения данных, что уменьшает время верификации и позволяет осуществить проверку динамических параметров микросхемы. Полученные результаты моделирования позволяют сократить ошибки при проектировании аппаратных средств, реализующих интерфейс связи микросхем памяти с процессором.

Список литературы

1. IS61LV6416L 64K x 16 High-Speed CMOS Static RAM With 3.3V Supply. [Electronic resource] / Интернет-ресурс. – Режим доступа: <http://www.datasheetarchive.com/IS61LV64>.
2. Single Port RAM Asynchronous Read/Write. [Electronic resource] / Интернет-ресурс. – Режим доступа: http://www.asic-world.com/examples/verilog/ram_sp_ar_aw.html.
3. Аль Мади М.К. Алгоритмы тестового диагностирования полупроводниковых запоминающих устройств: учеб. пособ. / Аль Мади М.К., Моамар Д.Н., Рябцев В.Г. – К.: "Корнійчук", 2008. – 220 с.
4. Рябцев В. Г. Метод и средство визуализации алгоритмов тестов диагностирования запоминающих устройств / В. Г. Рябцев, Д. Н. Моамар // Электронное моделирование. – 2010. – Т. 32. – № 3. – С. 43–52.