Федеральное государственное автономное образовательное учреждение  
высшего образования   
"Национальный исследовательский университет   
"Высшая школа экономики"

Московский институт электроники и математики им. А. Н. Тихонова

Департамент компьютерной инженерии

Отчёт

о выполнении практической работы № 4

Тема работы: «Высокоуровневое моделирование аппаратных проектов. Верификация HDL проектов. DPI, PLI/VPI»

по курсу «Высокоуровневое и имитационное моделирование цифровых систем»

Выполнили:

Власов Р. В. БИВ186

Сегида Т. О. БИВ186

Принял

асс. МИЭМ НИУ ВШЭ

Американов А. А.

Оценка:

Москва 2022 г.

Оглавление

[1. Основная часть 3](#_Toc101549297)

[1.1. Ознакомиться с материалами из руководств 3](#_Toc101549298)

[1.2. Проверить работоспособность ModelSim или установить QuestaSim 3](#_Toc101549299)

[1.3. Перейти в папку examples 3](#_Toc101549300)

[1.4. Проанализировать примеры использования DPI и PLI/VPI и выполнить примеры 3](#_Toc101549301)

[1.5. Выполнить пример examples\tutorials\systemverilog\dpi\_basic\ и прочитать README 3](#_Toc101549302)

[1.6. Ознакомиться с примером examples\systemverilog\dpi\dpivpiperf и отразить в отчете, чем отличаются подходы DPI и VPI/PLI 4](#_Toc101549303)

[1.7. Выполнить пример examples\c\_windows\dpi (или examples\systemverilog\dpi\simple\_calls) 4](#_Toc101549304)

[1.8. Выполнить пример examples\systemverilog\dpi\openarray 4](#_Toc101549305)

[1.9. Ознакомиться с примерами examples\systemverilog\dpi\packed\_types и examples\systemverilog\dpiunpacked\_types 4](#_Toc101549306)

[1.10. Ознакомиться с примером examples\systemverilog\dpi\checkpoint 5](#_Toc101549307)

[1.11. Ознакомиться с примером examples\systemverilog\dpi\cpackages 5](#_Toc101549308)

[1.12. Ознакомиться с примером examples\systemverilog\dpi\create\_sv\_dynarray 5](#_Toc101549309)

[2. Выводы 5](#_Toc101549310)

[3. Список литературы 5](#_Toc101549311)

1. Основная часть
   1. Ознакомиться с материалами из руководств

VPI – [Verilog Procedural Interface](https://en.wikipedia.org/wiki/Verilog_Procedural_Interface) – интерфейс, позволяющий связывать verilog-код с C-кодом – из C вызывать verilog-задачи, а из verilog C-функции.

PLI – [Program Language Interface](https://en.wikipedia.org/wiki/Verilog#Program_Language_Interface_(PLI)) – интерфейс, предшествовавший VPI, выполнявший те же функции.

DPI – [Direct Programming Interface](https://en.wikipedia.org/wiki/SystemVerilog_DPI) – аналог VPI/PLI для SystemVerilog, но с более широкими возможностями – дополнительно позволят вызывать Unix-функции из SystemVerilog, а также позволяет связывать SystemVerilog-код с таковым на C, C++, SystemC, как и другими.

* 1. Проверить работоспособность ModelSim или установить QuestaSim

Устанавливать что-то новое в ходе выполнения лабораторной работы не пришлось.

* 1. Перейти в папку examples

Скачанный с google-диска примеры расположены в lab\_4\examples.

* 1. Проанализировать примеры использования DPI и PLI/VPI и выполнить примеры

Основное внимание уделить папкам: c\_windows, systemverilog, tutorials\systemverilog, verilog.

* + 1. Выполнить пример examples\tutorials\systemverilog\dpi\_basic\ и прочитать README

<https://habr.com/ru/post/155201/> – про Makefile

<https://habr.com/ru/post/319466/> – есть про vsim и vopt

vopt – оптимизатор дизайна (синтезируемого устройства). Так как его у нас не оказалось, оптимизацию пропускаем.

Так как работаем из-под Windows, для запуска примера воспользовались скриптом windows.bat в папке с примером.

Чтобы запустить пример, пришлось убрать оптимизацию дизайна, так как у нас не установлен vopt. Новый код скрипта windows.bat:

vlib work

vlog test.sv -dpiheader dpi\_types.h foreign.c

vsim -i test -do "add wave light; view source"

Скрипт не запускает симуляцию сам. Чтобы запустить симуляцию вручную, необходимо выбрать ModelSim => Simulate => Run => Run -all.

Изображение выглядит как текст, монитор, снимок экрана

Автоматически созданное описание

1. Вейвформа симуляции примера dpi\_basic.

Объяснение последовательности цветов в вейвформе:

1. RED
   1. симуляция начинается с красного, потому что это первое значение в объявлении типа (traffic\_signal)
2. GREEN
   1. устанавливается функцией sv\_GreenLight из test.sv
3. YELLOW
   1. устанавливается функцией sv\_YellowLight из foreign.c
4. RED
   1. устанавливается функцией sv\_RedLight из foreign.c
5. GREEN
   1. устанавливается функцией sv\_GreenLight из test.sv
   2. на вейвформе не видно этого сигнала, но видно, что последний RED в конце симуляции меняется
   3. его не видно, потому что после установки сигнала на GREEN симуляция заканчивается
   4. если добавить ожидание после его установки, он появится на вейвформе, как и остальные
      1. Ознакомиться с примером examples\systemverilog\dpi\dpivpiperf и отразить в отчете, чем отличаются подходы DPI и VPI/PLI

В verilog нет int, но есть integer, пришлось заменить.

Для запуска примера пришлось написать свои скрипты (dpi.bat и vpi.bat), так как вложенный (run.do) никак не работал. Также, для компиляции примера понадобилось добавить опцию `-sv` (dpi.bat).

VPI/PLI требует больше кода, потому что требуется каждый раз все, написанное не в verilog, регистрировать.

Лучше DPI, потому что он создан с учетом ошибок и проблем VPI/PLI. Также verilog-код с использованием DPI выглядит как verilog-код, а C-код как C-код, что теряется при использовании VPI/PLI.

* + 1. Выполнить пример examples\c\_windows\dpi (или examples\systemverilog\dpi\simple\_calls)

Файл результатов – lab\_4\examples\c\_windows\dpi\results.txt – содержит лог выполнения симуляции.

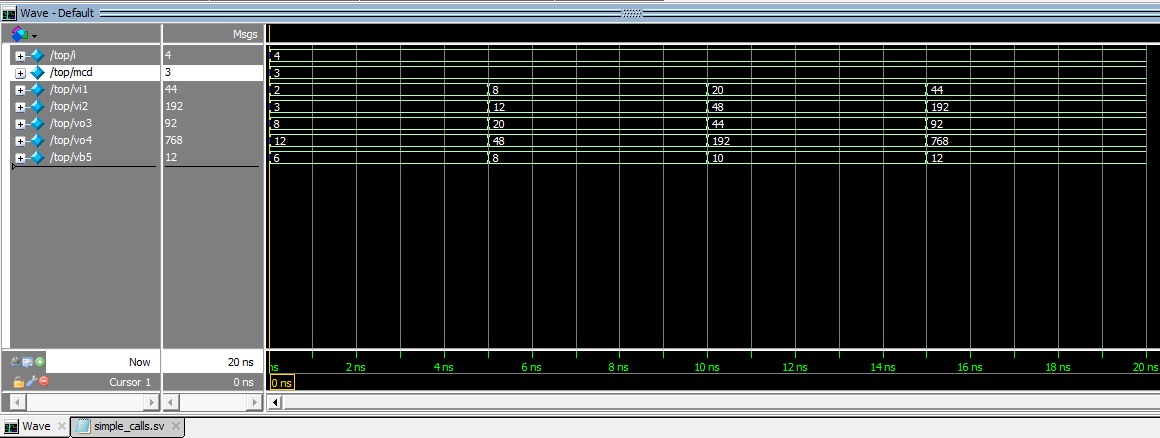
Чтобы запустить симуляцию не в режиме командной строки, а с графическим интерфейсом пришлось изменить скрипт vsim\_vs.do.

Заменили эту строку:

vsim -c top -do "run -all; quit -f"

На эту:

vsim -i top -do "add wave \*; run -all; view source"



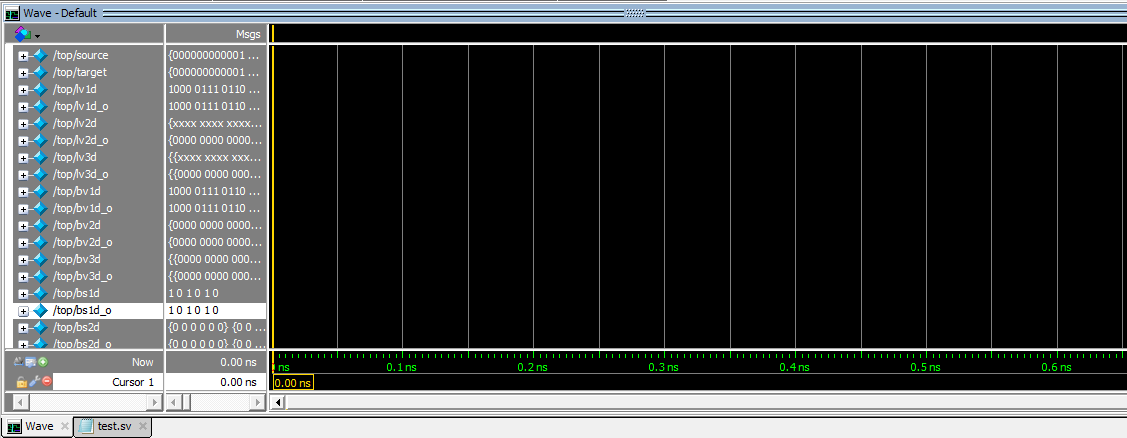
1. Вейвформа симуляции.

Verilog-код и C-код друг друга вызывают.

* + 1. Выполнить пример examples\systemverilog\dpi\openarray

В примере verilog-код и C-код передают друг другу массивы и копируют их.

Файл результатов – lab\_4\examples\systemverilog\dpi\openarray\results.txt.



1. Вейвформа симуляции.

Вейвформа пустая, потому что в тестбенче нет отсчета времени.

* + 1. Ознакомиться с примерами examples\systemverilog\dpi\packed\_types и examples\systemverilog\dpiunpacked\_types
       1. examples\systemverilog\dpi\packed\_types

Вейвформа снова пустая, по тем же причинам, что и в предыдущем примере.

Упакованные типы используются для экономии памяти.

* + - 1. examples\systemverilog\dpiunpacked\_types

Вейвформа снова пустая, по тем же причинам, что и в предыдущем примере.

При передаче неупакованных данных требуется дополнительно совершать операции упаковки и распаковки.

* + 1. Ознакомиться с примером examples\systemverilog\dpi\checkpoint

Данная технология может применяться для выполнения теста с ранее сохраненного этапа.

Может быть использована, например, для пропуска длительных по времени этапов, которые не изменяются от прогона к прогону.

* + 1. Ознакомиться с примером examples\systemverilog\dpi\cpackages

Данный прием может пригодиться для более тонкой настройки теста из verilog-кода. Например, чтобы при ошибках во время симуляции выполнять отличный код от того, что был бы без ошибок во время симуляции.

* + 1. Ознакомиться с примером examples\systemverilog\dpi\create\_sv\_dynarray

Данный прием может применяться для работы с массивами неизвестной начальной длины.

1. Выводы

В ходе работы был изучены технологии DPI и PLI/VPI и их отличия.

1. Список литературы
2. HLIMDS\_Lab\_5\_2020 [Электронный ресурс]. – URL: <https://docs.google.com/document/d/1vd78GVchQjIfopTo_ivaEGmXDZaFXkCFX8ypgCj-vOI/edit>
3. Интерфейс Verilog VPI [Электронный ресурс]. – URL: <https://marsohod.org/11-blog/266-verilog-vpi>
4. SystemVerilog DPI Tutorial [Электронный ресурс]. – URL: https://www.doulos.com/knowhow/systemverilog/systemverilog-tutorials/systemverilog-dpi-tutorial/