

# Практическое задание №1.

## *Структурный анализ схем.*

### **Общее описание задания**

Задание представляет первую часть составного задания по анализу и классификации логических схем. Основная цель указанного задания – реализация алгоритмов извлечения структурных параметров для заданной логической схемы. При выполнении задания можно выделить следующие основные этапы:

1. реализация структур данных для хранения структуры логической схемы;
2. реализация транслятора, позволяющего преобразовывать схемы, записанные в формате Verilog, в реализованные на предыдущем этапе структуры данных;
3. выбор множества анализируемых структурных параметров и реализация алгоритмов их расчета.

Задание выполнять в папке Spring 2015 – Homework 1 в Вашей папке именной папке в Dropbox. Так как задание выполняется в паре, то папка должна быть разделяемой между всеми исполнителями задания. Задание должно быть выполнено с использованием системы контроля версий Git (фактически, в этой папке Вы должны в самом начале создать Git репозиторий). Работа каждого студента будет оцениваться в первую очередь по тем commit-ам, которые он сделал. Так как это Ваш первый опыт парной работы с использованием системы контроля версий, рекомендуется, чтобы у Вас была локальная копия сделанной работы. Это позволит минимизировать риск потери работы при несогласованной работе с системой контроля версий.

Распределение студентов по группам будет размещено на сайте кафедры.

Для тестирования результатов выполнения практического задания на сайте кафедры будетложен набор тестовых схем. Кроме того, программа будет тестироваться на скрытом наборе схем.

Результаты выполнения практического задания сопровождаются отчетом о проделанной работе, подробной инструкцией по сборке и запуске программы и makefile-ом.

Любые вопросы по заданию присыпать по электронной почте на следующий адрес: [mikle.shupletsov@gmail.com](mailto:mikle.shupletsov@gmail.com).

Тема письма имеет следующий формат: [318] [Фамилия Имя] [Вопрос].

### **Реализация структур данных для хранения структуры логической схемы.**

Требуется реализовать согласованную структуру классов, которая позволит хранить граф рассматриваемой схемы и выполнять необходимые операции для вычисления выбранного набора структурных параметров. Для того, чтобы ускорить разработку, предполагается, что для реализации основной функциональности классов будет использоваться библиотека Boost Graph Library ([http://www.boost.org/doc/libs/1\\_57\\_0/libs/graph/doc/index.html](http://www.boost.org/doc/libs/1_57_0/libs/graph/doc/index.html)). Для того, чтобы объем репозитория на Dropbox был разумным, библиотеку загружать в репозиторий

запрещается. Библиотека должна быть загружена на локальную машину и соответствующим образом прописана в путь.

## Реализация транслятора с языка Verilog.

Требуется реализовать транслятор схем из языка Verilog в разработанные на предыдущем этапе структуры данных. При этом считается, что схемы заданы так, что использует довольно небольшое подмножество конструкций языка Verilog. На схему накладываются следующие ограничения:

1. схема комбинационная (не содержит регистров);
2. схема состоит из одного модуля с именем top;
3. схема представлена на уровне функциональных элементов (gate-level) и при этом используются только стандартные примитивы языка Verilog (buf, not, and, nand, or, nor, xor, xnor), которые могут иметь произвольное конечное число входов (кроме buf и not);
4. все элементы соединяются только при помощи wire (соединения могут быть как скалярные, так и векторные).

*Пример входного файла на языке Verilog:*

```
module top(a, sel, out);
    input [2:0] a;
    input      sel;
    output [1:0] out;
    wire [2:0] a;
    wire      sel;
    wire [1:0] out;
    wire [3:0] z;
    wire w1, w2, w3, w4, w5, w6, w7;
    not (z[0], a[0]);
    xnor (z[1], a[0], a[1]);
    or  (w1, a[0], a[1]);
    xnor (z[2], a[2], w1);
    and (w2, a[2], a[1]);
    and (w3, a[2], a[0]);
    nor (z[3], w2, w3);
    nor (w4, sel, z[3]);
    nor (w5, sel, z[2]);
    and (w6, sel, z[1]);
    and (w7, sel, z[0]);
    or (out[1], w4, w6);
    or (out[0], w5, w7);
endmodule
```

Функциональность реализованного транслятора должна быть достаточной для того, чтобы обрабатывать логические схемы, удовлетворяющие указанным ограничениям. Другие конструкции языка Verilog реализовывать не требуется.

## Структурные параметры схемы.

Предполагается, что в рамках задания каждая группа реализует алгоритмы расчета выбранного ими набора структурных параметров логической схемы. Структурными

параметрами считаются любые параметры, связанные со структурой графа схемы. Например, параметры вершин и ребер схемы, структура путей и подсхем в схеме и т.д. При этом этот набор должен включать в себя как ряд обязательных параметров (у каждой группы набор обязательных параметров определяется индивидуально), так и ряд параметров по выбору группы. При этом поощряется реализация алгоритмов расчета дополнительных параметров, которая будет оцениваться отдельно.

### *Распределение обязательных параметров по группам:*

1. Группа №1:
  - а. средняя степень вершин в графе схемы;
  - б. спектр длин путей в графе схемы (для каждого значения длины пути в схеме (от минимального до максимального значения) вычисляется доля путей (рассматриваются только пути от входов до выхода схемы), длина которых равна указанному значению);
  - с. среднее число вершин во входном конусе вершины в графе схемы;
2. Группа №2:
  - а. спектр степеней вершин (для каждого значения степени вершины (от минимального до максимального значения) вычисляется доля вершин, степень которых равна указанному значению);
  - б. средняя длина пути в графе схемы (от входов до выхода схемы)
  - с. среднее число входных переменных во входном конусе вершины в графе схемы;
3. Группа №3:
  - а. средняя полустепень захода вершин в графе схемы;
  - б. средняя длина пути в графе схемы (по всем вершинам в графе схемы);
  - с. спектр числа вершин во входном конусе вершины в графе схемы (для каждого значения числа вершин во входном конусе вершины (от минимального до максимального значения) вычисляется доля вершин, для которых число вершин в их входном конусе равно указанному значению);
4. Группа №4:
  - а. спектр полустепеней захода вершин (для каждого значения полустепени захода вершины (от минимального до максимального значения) вычисляется доля вершин, степень которых равна указанному значению);
  - б. максимальная длина пути в графе схемы (от входов до выхода схемы);
  - с. средняя степень вершин во входном конусе вершины в графе схемы;
5. Группа №5:
  - а. минимальная степень вершин в графе схемы;
  - б. спектр длин путей в графе схемы (для каждого значения длины пути в схеме (от минимального до максимального значения) вычисляется доля путей (рассматриваются только пути от входов до выхода схемы), длина которых равна указанному значению);
  - с. средняя полустепень захода во входном конусе вершины в графе схемы;
6. Группа №6:
  - а. спектр полустепеней исхода вершин (для каждого значения полустепени захода вершины (от минимального до максимального значения) вычисляется доля вершин, степень которых равна указанному значению);
  - б. минимальная и максимальная длина пути в графе схемы (от входов до выхода схемы);
  - с. спектр числа входных переменных во входном конусе вершины в графе схемы (для каждого значения числа входных переменных во входном конусе вершины (от минимального до максимального значения)

вычисляется доля вершин, для которых число входных переменных в их входном конусе равно указанному значению);

Каждая группа может дополнительно реализовать расчет любого количества параметров, представленных в Приложении 1, а также предложить и реализовать расчет параметров, которые не представлены в указанном приложении. Реализация расчета дополнительных параметров будет оцениваться отдельно.

## *Приложение 1.*

1. средняя степень вершин в графе схемы;
2. средняя полустепень захода вершин в графе схемы;
3. средняя полустепень исхода вершин в графе схемы;
4. минимальная степень вершин в графе схемы;
5. максимальная степень вершин в графе схемы;
6. спектр степеней вершин (для каждого значения степени вершины (от минимального до максимального значения) вычисляется доля вершин, степень которых равна указанному значению);
7. спектр полустепеней захода вершин (для каждого значения полустепени захода вершины (от минимального до максимального значения) вычисляется доля вершин, степень которых равна указанному значению);
8. спектр полустепеней исхода вершин (для каждого значения полустепени захода вершины (от минимального до максимального значения) вычисляется доля вершин, степень которых равна указанному значению);
9. средняя длина пути в графе схемы (по всем вершинам в графе схемы);
10. средняя длина пути в графе схемы (от входов до выхода схемы)
11. минимальная длина пути в графе схемы (от входов до выхода схемы);
12. максимальная длина пути в графе схемы (от входов до выхода схемы);
13. спектр длин путей в графе схемы (для каждого значения длины пути в схеме от минимального до максимального значения вычисляется доля путей (рассматриваются все возможные пути в графе схемы), длина которых равна указанному значению);
14. спектр длин путей в графе схемы (для каждого значения длины пути в схеме (от минимального до максимального значения) вычисляется доля путей (рассматриваются только пути от входов до выхода схемы), длина которых равна указанному значению);
15. среднее число вершин во входном конусе вершины в графе схемы;
16. среднее число входных переменных во входном конусе вершины в графе схемы;
17. средняя степень вершин во входном конусе вершины в графе схемы;
18. средняя полустепень захода во входном конусе вершины в графе схемы;
19. средняя полустепень исхода во входном конусе вершины в графе схемы;
20. спектр числа вершин во входном конусе вершины в графе схемы (для каждого значения числа вершин во входном конусе вершины (от минимального до максимального значения) вычисляется доля вершин, для которых число вершин в их входном конусе равно указанному значению);
21. спектр числа входных переменных во входном конусе вершины в графе схемы (для каждого значения числа входных переменных во входном конусе вершины (от минимального до максимального значения) вычисляется доля вершин, для которых число входных переменных в их входном конусе равно указанному значению);