

КМ-3. Последовательностные схемы

Задание 1

1. На основе представлений для динамического D -триггера, сформировать структурное описание JK - и T -триггера, произвести RTL-синтез.
2. Для JK - и T -триггера привести поведенческое описание модели, произвести RTL-синтез.
3. Сравнить RTL-представление для п. 1 и 2 по количеству используемых логических элементов и триггеров (отчёт о синтезе в среде Quartus Prime).
4. Написать общий testbench, продемонстрировать идентичность результатов моделирования.

Задание 2

1. Используя параметризацию спроектировать один модуль, осуществляющий логический сдвиг влево и вправо, а также арифметический сдвиг вправо N -разрядного числа на M разрядов с использованием операторов сдвига.
2. Используя параметризацию спроектировать один модуль, осуществляющий логический сдвиг влево и вправо, а также арифметический сдвиг вправо N -разрядного числа на M разрядов с последовательной логики.
3. Написать для каждого модуля testbench, продемонстрировать идентичность результатов моделирования.

Указание:

Для чётных номеров по списку $N = 32$, для нечётных $N = 64$. Количество разрядов, на которое сдвигается число равно номеру варианта.

Задание 3

Спроектировать реверсивный счётчик с коэффициентом счёта K с возможностью задания начального значения. Продемонстрировать режимы работы.

Вариант	Коэффициент счёта K
1, 6, 11, 16, 21	10
2, 7, 12, 17, 22	20
3, 8, 13, 18, 23	13
4, 9, 14, 19, 24	8
5, 10, 15, 20, 25	16

Указание:

Для ОС Windows и семейства Linux моделирование для ускорения можно производить при помощи пакетов Icarus Verilog (iVerilog) и GTKWave <https://marsohod.org/11-blog/113-icarus>. Стоит отметить, что при моделировании иерархических файлов типа testbench

необходимо в команде для iVerilog в обратном порядке записать все файлы иерархии, например, для рассмотренного на занятии D-триггера D flip flop testbench запишется как

```
> iverilog -o qq RS.v D_latch.v D_flip_flop.v tb_D_flip_flop.v
```

Ключ **-o** со значением **qq** означает, что все выходные данные по результатам моделирования будут расположены в файле **qq**. Если этот ключ отсутствует, то по умолчанию будет создаваться файл **a.out**.

Чтобы избежать такой необходимости «помнить» все иерархические вложения, можно воспользоваться директивой **`include «<filename.v>»**, которая в нужном файле подключает файл нижнего уровня иерархии. Так, например, в **D_latch.v** нам необходимо использование RS-триггера, который описан в модуле **RS**, расположенном в файле **RS.v**. Соответственно, перед описанием модуля **D_latch** в файле **D_latch.v** необходимо прописать **`include «RS.v»**.

Если в testbench прописана команда **\$dumpvars**, то по выполнению её iVerilog создаст файл **dump.vcd**, в котором будет храниться изменения всех сигналов. Для формирования этого файла необходимо запустить демонстрацию результатов моделирования командой

```
> vvp qq
```

В виде осциллограммы этот файл можно воспроизвести в пакете GTKWave командой

```
> gtkwave dump.vcd
```

NB: при описании в программе Quartus Prime или ModelSim, где удобнее создать проект со всеми файлами, в препроцессорной директиве **`include** нет необходимости.