

# ОСНОВЫ VerilogHDL/SystemVerilog (синтез и моделирование)

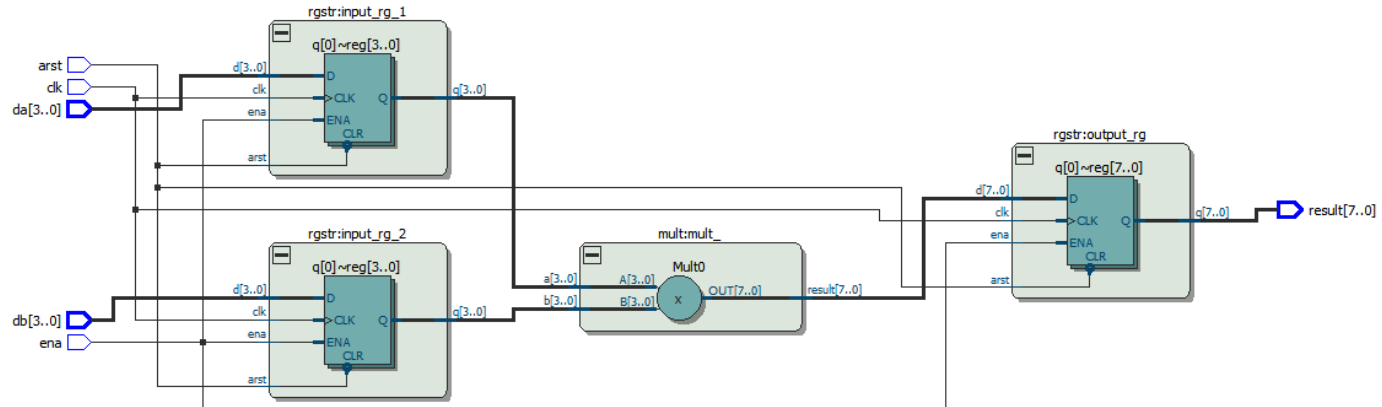
**Задание lab5 (4 часа лабораторных занятий)**

# Часть lab5\_1

## Часть lab5\_1

- На языке Verilog разработать параметризированный модуль, который содержит:

- ✓ 2 входных регистра; Умножитель; Выходной регистр
- ✓ ena – (key[0]) – разрешение работы (при 1)
- ✓ arst - (key[1]) – асинхронный сброс (при 0)
- ✓ clk - тактовый сигнал
- ✓ da[3..0] – sw[7..4],
- ✓ db[3..0] – sw[3..0]
- ✓ result [7..0] – led [7..0]



- Необходимо создать:

- ✓ Параметризированный регистр
- ✓ Параметризированный умножитель
- ✓ Параметризированное описание верхнего уровня

- Имя проекта – lab5\_1. Имя модуля верхнего уровня –lab5\_1.

# Часть lab5\_1 (Порядок выполнения)

---

- ❑ Создать проект:
  - ✓ Имя проекта – lab5\_1
- ❑ Создать параметризируемый модуль регистра (базовое значение параметра – 8), посмотреть синтезированную пакетом схему, осуществить его моделирование
- ❑ Создать параметризированный модуль умножителя (базовое значение параметра – 8), посмотреть синтезированную пакетом схему, осуществить его моделирование
- ❑ Создать структурное описание верхнего уровня - имя модуля верхнего уровня иерархии lab5\_1(описание должно быть параметризированным, базовое значение параметра - 4),
- ❑ Посмотреть синтезированную пакетом схему (RTL Viewer) имя модуля верхнего уровня иерархии
- ❑ Осуществить моделирование имя модуля верхнего уровня иерархии
- ❑ Назначить выводы СБИС с помощью атрибутов,
- ❑ Осуществить компиляцию и проверить работу на плате.

# Часть lab5\_1 Пример: Регистр, умножитель и модуль верхнего уровня

```
module rgstr (arst, ena, clk, d, q);
parameter width_rg = 8;

input clk, arst, ena;
input [width_rg-1:0] d;
output reg [width_rg-1:0] q;

always @(posedge clk, negedge arst)
if (arst==1'b0) q <= {width_rg{1'b0}};
else if (ena) q <= d;

endmodule
```

```
module mult (a, b, result);
parameter width_mult = 8;

input [width_mult-1:0] a, b;
output [2*width_mult-1:0] result;

assign result = a*b;

endmodule
```

```
module lab9 (arst, ena, clk, da, db, result);
parameter width = 4;

input clk, arst, ena;
input [width-1:0] da, db;
output [2*width-1:0] result;

wire [width-1:0] da_int, db_int;
wire [2*width-1:0] mult_res;

rgstr #width input_rg_1 (arst, ena, clk, da, da_int );
rgstr #width input_rg_2 (.arst(arst),
                        .ena(ena), .clk(clk),
                        .d(db),
                        .q (db_int) );

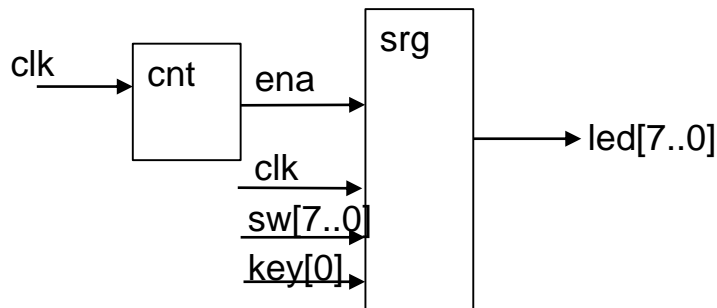
mult #width mult_ (da_int, db_int, mult_res);

rgstr #(2*width) output_rg (arst, ena, clk, mult_res, result );

endmodule
```

# Часть lab5\_2

- На языке Verilog создать иерархическое (структурное) описание представленной ниже схемы



Ena – выход счетчика-делителя  
разрешающий работу  
сдвигающему регистру

- ✓ cnt – параметризованный счетчик-делитель: коэффициент деления задается параметром; выход – сигнал переноса
- ✓ srg – параметризованный кольцевой сдвигающий регистр с параллельной загрузкой, описание которого выполнено с помощью цикла **for loop**
- ✓ Входы:
  - Clk – тактовый (25 МГц), sw[7..0] – данные загружаемые в сдвигающий регистр, key[0] – сигнал **синхронной** загрузки данных в сдвигающий регистр
- ✓ Выход – led[7..0] – содержимое сдвигающего регистра

- Имя проекта – lab5\_2. Имя модуля верхнего уровня –lab5\_2.

# Часть lab5\_2 (Порядок выполнения)

---

- ❑ Создать проект:
  - ✓ Имя проекта – lab5\_2,
- ❑ Создать параметризируемый модуль счетчика делителя (базовое значение параметра – 3), посмотреть синтезированную пакетом схему (RTL Viewer) , осуществить его моделирование
- ❑ Создать параметризированный модуль кольцевого сдвигающего регистра с параллельной загрузкой (базовое значение параметра – 8), посмотреть синтезированную пакетом схему (RTL Viewer) , осуществить его моделирование
- ❑ Создать структурное описание верхнего уровня , имя модуля верхнего уровня иерархии lab5\_2, (описание должно быть параметризированным: задаются параметры:
  - ✓ деления частоты для счетчика делителя (базовое значение параметра - 4),
  - ✓ разрядность кольцевого сдвигающего регистра (базовое значение параметра - 8),
- ❑ Посмотреть синтезированную пакетом схему (RTL Viewer)
- ❑ Осуществить моделирование
- ❑ Назначить значение параметра счетчика делителя для реализации на плате (25 000 000)
- ❑ Назначить выводы СБИС с помощью атрибутов,
- ❑ Осуществить компиляцию и проверить работу на плате.

# Часть lab5\_3

---

- ❑ На языке Verilog **на структурном уровне** создайте иерархический проект, содержащий:
- ❑ На верхнем уровне: мультиплексор  $2(4\text{-разрядных входов}) \Rightarrow 1(4\text{-разрядный выход})$ ,
  - ✓ Состоит из мультиплексоров  $mux2\_1$  (мультиплексор  $2(1\text{-разрядных входов}) \Rightarrow 1(1\text{-разрядный выход})$ )
- ❑ Мультиплексор  $mux2\_1$  опишите на структурном уровне с использованием **Gate-Level примитивов языка Verilog**.
- ❑ Выводы устройства:
  - ✓ Входы:
    - Переключатели  $sw[3:0]$  – вход  $ina$
    - Переключатели  $sw[7:4]$  – вход  $inb$
    - Кнопка  $rba$  – вход управления мультиплексором (кнопка нажата – на выход передается значение с входа  $ina$ , кнопка не нажата -  $inb$ ).
  - ✓ Выходы
    - $led[3:0]$  подключены к светодиодам.
- ❑ Имя проекта – `lab5_3`. Имя модуля верхнего уровня – `lab5_3`.

# Часть lab5\_3 (Порядок выполнения)

---

- ❑ Создать проект:
  - ✓ Имя проекта – lab5\_3,
- ❑ Создайте на структурном уровне с использованием **Gate-Level примитивов языка Verilog** описание мультиплексора mux2\_1 (мультиплексор 2(1-разрядных входа)=>1 (1-разрядный выход))
- ❑ Посмотрите синтезированную пакетом схему (RTL Viewer) , осуществите его моделирование
- ❑ **Создайте структурное описание верхнего уровня** - мультиплексор 2(4-разрядных входа)=>1 (4-разрядный выход), состоящий из мультиплексоров mux2\_1, имя модуля верхнего уровня иерархии lab5\_3.
- ❑ Посмотрите синтезированную пакетом схему (RTL Viewer)
- ❑ Осуществите моделирование
- ❑ Назначьте **выводы СБИС с помощью атрибутов**,
- ❑ Осуществите компиляцию и проверьте работу на плате.