

# Языки описания схем

(mk.cs.msu.ru → Лекционные курсы → Языки описания схем)

## Блок ПЗ

Практика:  
Verilog и процедуры

Лектор:  
**Подымов Владислав Васильевич**

E-mail:  
**valdus@yandex.ru**

# Упражнение 1

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля с перебором всевозможных наборов булевых значений на входах

## 1. Демультиплексор с двумя однобитовыми выходами

входы ширины 1:  $x$ ,  $s$

выходы ширины 1:  $y_0$ ,  $y_1$ :

- ▶  $y_s = x$
- ▶  $y_{1-s} = 0$

## 2. Демультиплексор с двумя двухбитовыми выходами

входы     ▶ ширины 2:  $x$

           ▶ ширины 1:  $s$

выходы ширины 2:  $y_0$ ,  $y_1$ :

- ▶  $y_s = x$
- ▶  $y_{1-s} = 0$

# Упражнение 1

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля с перебором всевозможных наборов булевых значений на входах

## 3. Демультиплексор с четырьмя однобитовыми выходами, собранными в шину

- входы
- ▶ ширины 1:  $x$
  - ▶ ширины 2:  $s$

выход ширины 4:  $y$ :

- ▶  $s$ -й разряд выхода  $y$  равен  $x$
- ▶ значение в остальных разрядах — 0

# Упражнение 1

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля с перебором всевозможных наборов булевых значений на входах

## 4. Простое двухразрядное арифметическое устройство

**входы**

- ▶ ширины 2:  $x, y$
- ▶ ширины 1:  $s$

**выход** ширины 4:  $z$ :

- ▶ если  $s = 0$ , то  $z = x + y$
- ▶ если  $s = 1$ , то  $z = x \cdot y$

## 5. Простое трёхразрядное арифметическое устройство

**входы**

- ▶ ширины 3:  $x, y$
- ▶ ширины 1:  $s$

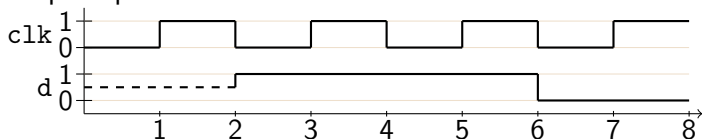
**выход** ширины 6:  $z$ :

- ▶ если  $s = 0$ , то  $z = x + y$
- ▶ если  $s = 1$ , то  $z = x \cdot y$

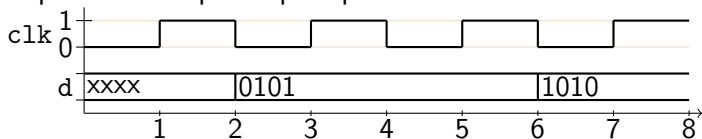
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 1. D-триггер



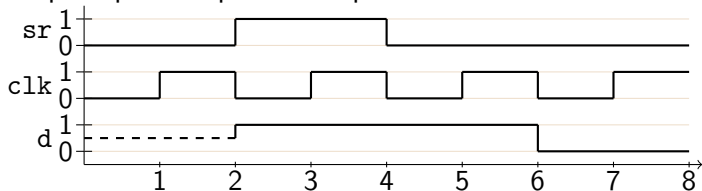
### 2. Параллельный регистр ширины 4



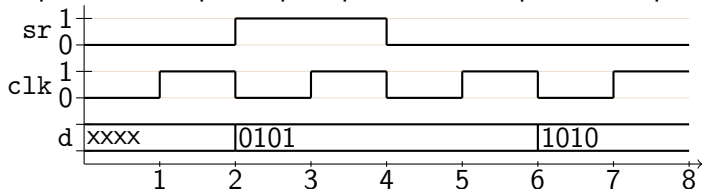
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 3. D-триггер с синхронным сбросом



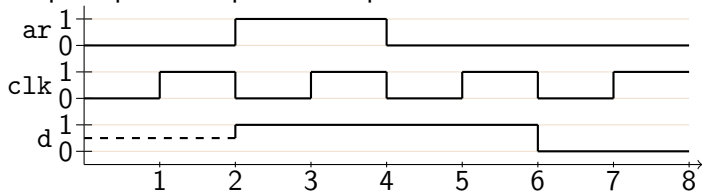
### 4. Параллельный регистр ширины 4 с синхронным сбросом



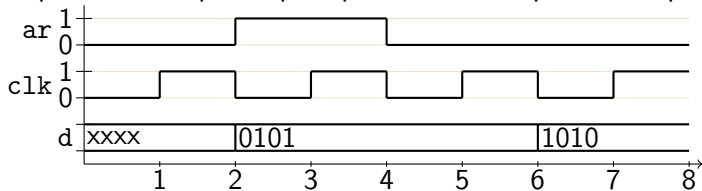
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 5. D-триггер с асинхронным сбросом



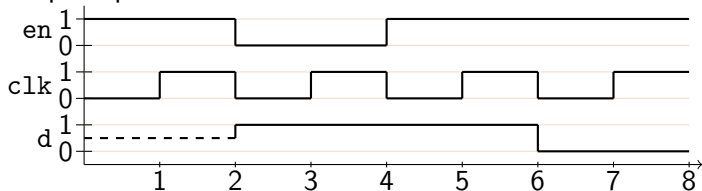
### 6. Параллельный регистр ширины 4 с асинхронным сбросом



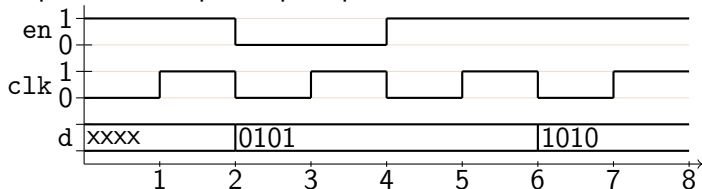
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 7. D-триггер с включением



### 8. Параллельный регистр ширины 4 с включением

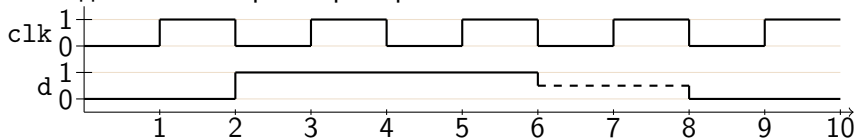




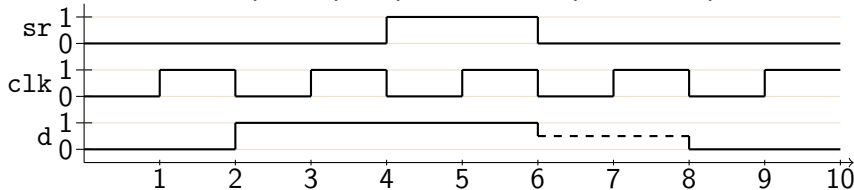
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 9. Последовательный регистр ширины 4



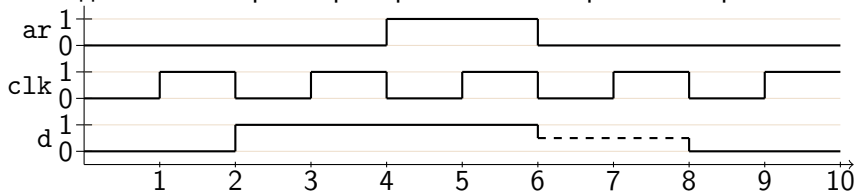
### 10. Последовательный регистр ширины 4 с синхронным сбросом



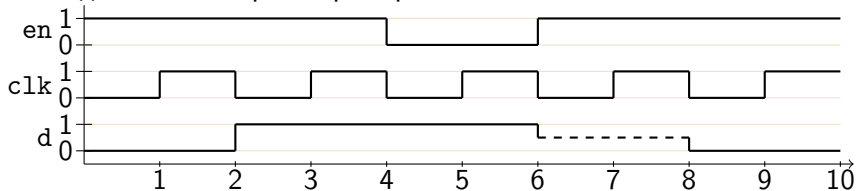
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 11. Последовательный регистр ширины 4 с асинхронным сбросом



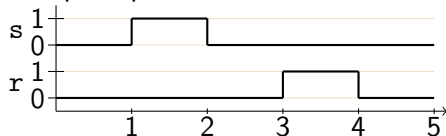
### 12. Последовательный регистр ширины 4 с включением



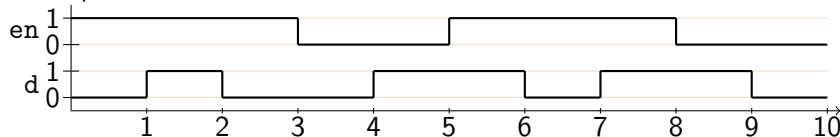
## Упражнение 2

- ▶ Разработать модуль с заданным поведением в поддерживаемом фрагменте синтаксиса языка Verilog
- ▶ Выполнить программную симуляцию модуля для заданного сценария выполнения

### 13. RS-триггер



### 14. D-защёлка



## Упражнение 3

Разработать модуль **синхронной** последовательной схемы с тактовым входом *clk*, входом асинхронного сброса *ar* и заданным поведением в поддерживаемом фрагменте языка Verilog:

### 1. Счётчик чётности

**вход** ширины 1:  $x$

**выход** ширины 1:  $y$ :

- ▶  $y(0) = 0$
- ▶ если  $x(t) = 0$ , то  $y(t + 1) = y(t)$ , иначе  $y(t + 1) = \overline{y(t)}$

### 2. Двухбитовый счётчик

**выход** ширины 2:  $y$ :

- ▶ порядок следования чисел:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$
- ▶  $y(0) = 0$
- ▶  $y(t) \rightarrow y(t + 1)$

## Упражнение 3

Разработать модуль **синхронной** последовательной схемы с тактовым входом *clk*, входом асинхронного сброса *ar* и заданным поведением в поддерживаемом фрагменте языка Verilog:

### 3. Хаотичный двухбитовый счётчик

**выход** ширины 2: *y*:

- ▶ порядок следования чисел:  $0 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 0$
- ▶  $y(0) = 0$
- ▶  $y(t) \rightarrow y(t + 1)$

### 4. Условно-хаотичный двухбитовый счётчик

**вход** ширины 1: *x*

**выход** ширины 2: *y*:

- ▶  $y(0) = 0$
- ▶ если  $x = 0$ , то  $y(t) \rightarrow y(t + 1)$  согласно двухбитовому счётчику
- ▶ если  $x = 1$ , то  $y(t) \rightarrow y(t + 1)$  согласно хаотичному двухбитовому счётчику

## Упражнение 3

Разработать модуль **синхронной** последовательной схемы с тактовым входом *clk*, входом асинхронного сброса *ar* и заданным поведением в поддерживаемом фрагменте языка Verilog:

### 5. Таймер

**ВЫХОД** ширины 1: *y*:

$$\blacktriangleright y(t) = 1 \Leftrightarrow t \geq 3$$

### 6. Распознаватель 1

**ВХОД** ширины 1: *x*

**ВЫХОД** ширины 1: *y*:

$$\blacktriangleright y(t) = 1 \Leftrightarrow t \geq 4 \text{ и } (x(t-3)x(t-2)x(t-1)) = (101)$$

### 7. Распознаватель 2

**ВХОД** ширины 1: *x*

**ВЫХОД** ширины 1: *y*:

$$\blacktriangleright y(t) = 1 \Leftrightarrow \exists t' : 4 \leq t' \leq t \text{ и } (x(t'-3)x(t'-2)x(t'-1)) = (101)$$

## Упражнение 3

Разработать модуль **синхронной** последовательной схемы с тактовым входом *clk*, входом асинхронного сброса *ar* и заданным поведением в поддерживаемом фрагменте языка Verilog:

### 8. Делитель частоты 1

выход ширины 1: *y*:

- ▶ *y* — тактовый сигнал вдвое меньшей частоты по сравнению с *clk*

### 9. Делитель частоты 2

выход ширины 1: *y*:

- ▶ *y* — тактовый сигнал вчетверо меньшей частоты по сравнению с *clk*

### 10. Делитель частоты 3

выход ширины 1: *y*:

- ▶ *y* — тактовый сигнал втрое меньшей частоты по сравнению с *clk*