

**AQUARIUS**

**Самостоятельная 1.  
Терминал.  
Симуляция.**

**AQUARIUS**

# **Часть 1. Работа в основной ОС.**

# Тестовое окружение

```
module tb;
```

```
logic[3:0] result;  
logic clk;  
logic a;  
logic b;
```

Объявление  
проводов

```
//  
top i_top  
(  
    .a      ( a ),  
    .b      ( b ),  
    .result ( result )  
);
```

Объявление экземпляра  
модуля и подключение  
проводов к нему

```
//  
initial  
begin  
    $dumpvars;  
    repeat (8)  
    begin  
        # 10  
        a <= $urandom ();  
        b <= $urandom ();  
    end  
    $finish;  
end
```

Блок создания  
воздействий

Создание файла  
временных диаграмм  
(dump.vcd)

Задержка на  
10 тактов  
симуляции

Применение  
случайных  
воздействий

Окончание  
симуляции

```
endmodule
```

# Дешифратор 3в8

- Создайте директорию `mkdir -p ~/practice/my_projects/decoder`
- Запустите `VS code` и создайте файл `top.sv` в директории `decoder`
- Опишите декодер 3в8 при помощи

*\*обратите внимание на системную функцию `$dumpvars` в файле `tb.sv`. Она позволяет создавать файл `dump.vcd`*

а) assign и логический сдвиг	б) always @(*) + if-else	в) always @(*) + case	г) always @(*) + ?:	д) assign, ~, &
------------------------------	--------------------------	-----------------------	---------------------	-----------------

- Создайте файл `tb.sv`, производящий на входы устройства:

а) Случайные воздействия	б) Последовательный перебор при помощи цикла: <code>for (integer data_in=0; data_in&lt;=3'b111; data_in++)</code>
--------------------------	--

- Скомпилируйте Verilog командой `iverilog -g2012 -s tb top.sv tb.sv`
- Запустите файл `a.out` командой `vvp a.out`

*\*На текущий момент Icarus не поддерживает симуляцию массивов шин*

- Запустите программу просмотра временных диаграмм `gtkwave dump.vcd`
- \*Реализуйте функцию проверки логики устройства с выводом результата в консоль

# D-триггер

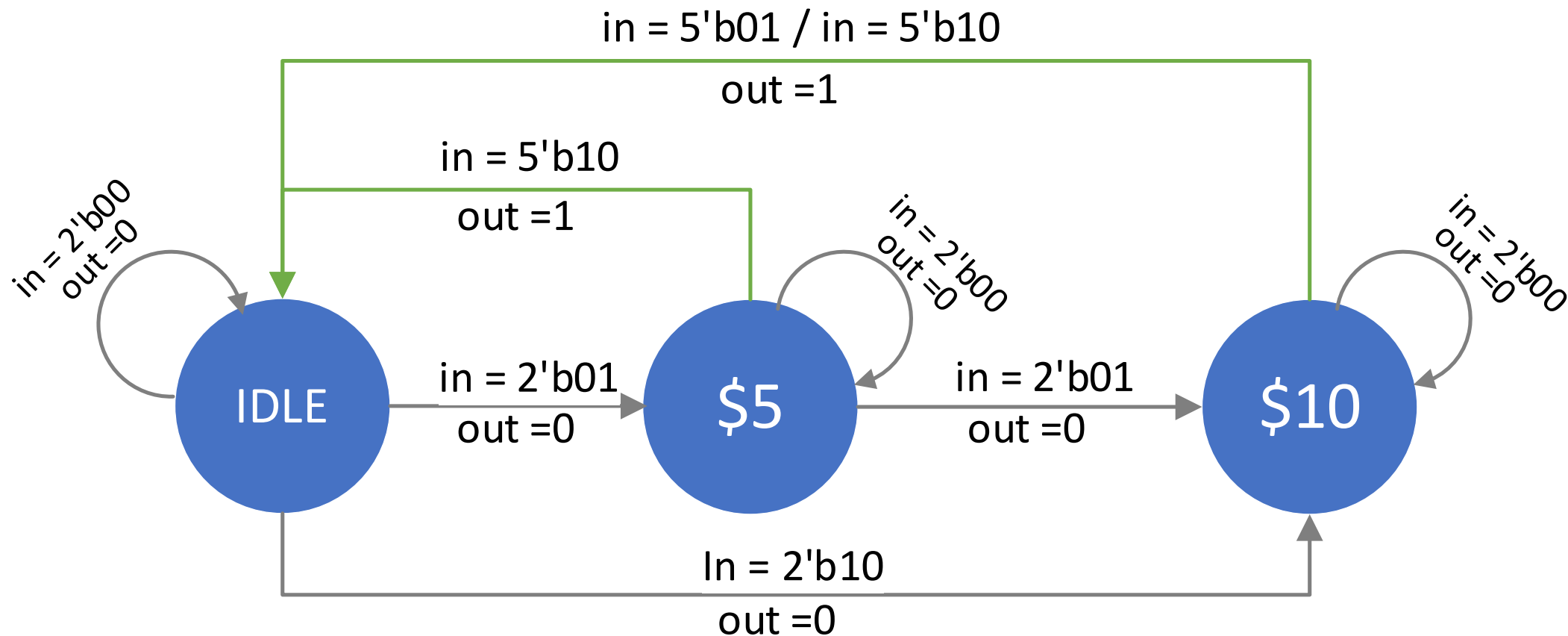
1. Создать директорию flip-flop в папке ~/practice/my\_projects командой  
`mkdir -p ~/practice/my_projects/flip-flop`
2. Описать D-триггер в файле flip\_flop.sv
3. Создать файл tb.sv, произвести случайное воздействие на входы.
4. Произвести симуляцию. `iverilog -g2012 -s tb tb.sv flip_flop.sv` - запуск симуляции
5. Рассмотреть временные диаграммы.

**AQUARIUS**

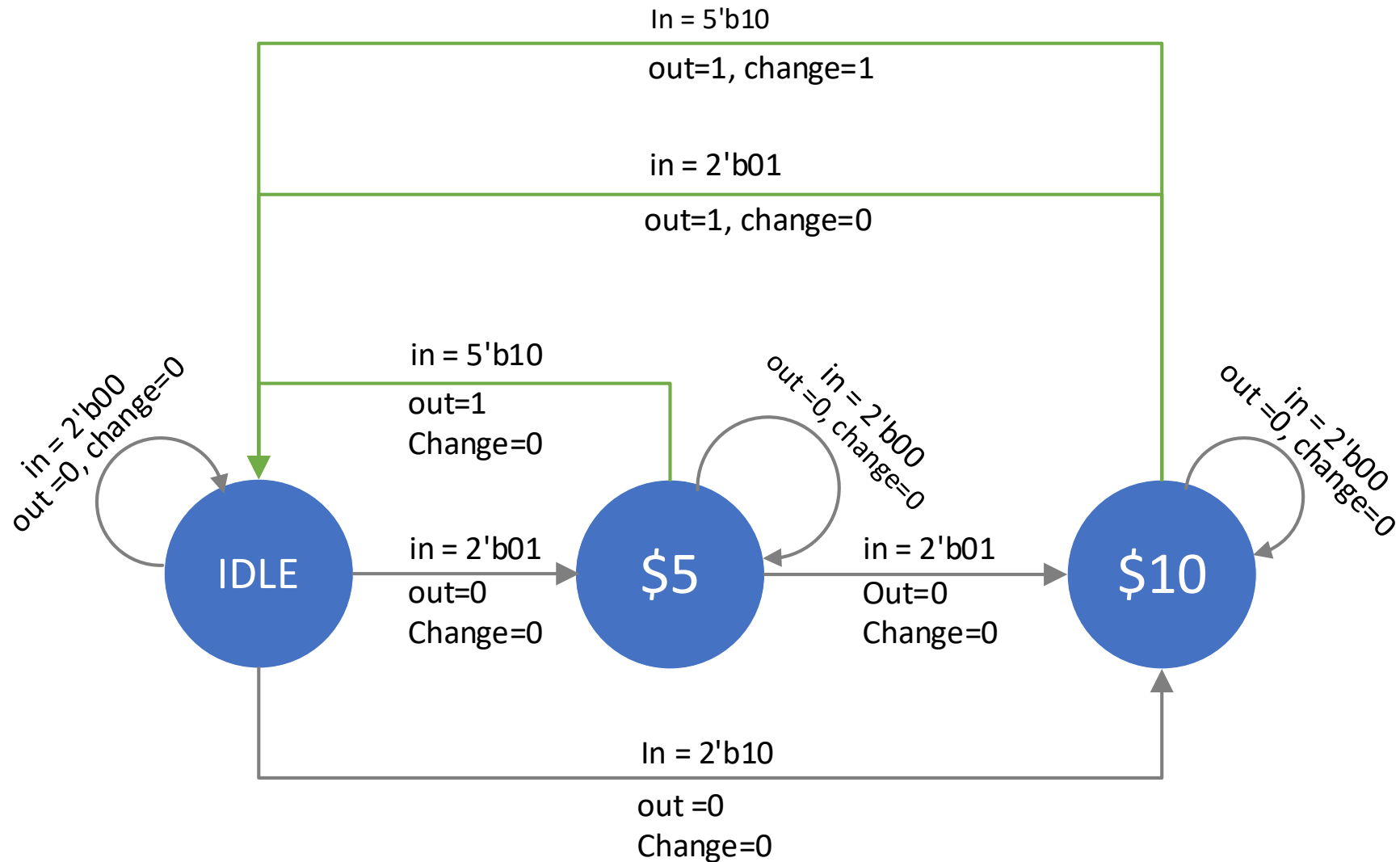
## **Часть 2. Работа на виртуальной машине**

# Пример конечного автомата (Торговый автомат без сдачи)

Автомат продает товар по цене 15\$, но не выдает сдачу



# Задача: добавьте возможность выдачи сдачи





# Указания по выполнению

1. Перейдите в директорию ~/practise/fsm\_lab.
2. Ознакомьтесь с кодом.
3. Проведите компиляцию командой **vlog vending\_machine.sv tb.sv**
4. Запустите ModelSim командой **vsim -voptargs="+acc" tb**
5. Добавьте необходимые сигналы для просмотра состояний (c\_state, n\_state)
6. Чтобы добавить все сигналы используйте **add wave -recursive -depth 10 \*** или навигацию с помощью мыши
7. Запустите симуляцию командой **run -all**
8. Are you sure you want to finish? – **No**
9. Повторный запуск осуществляется командой **restart**
10. Для запуска в консольном режиме из терминала используйте команду **vsim -c tb**

# Перечень полезных ссылок

<https://docstech.ru/icvgc-startwork/> - гайд по началу работы с icarus. На русском языке.

<https://steveicarus.github.io/iverilog/index.html> - англоязычная документация по работе с icarus.

[https://steveicarus.github.io/iverilog/usage/command\\_line\\_flags.html](https://steveicarus.github.io/iverilog/usage/command_line_flags.html) - описание флагов icarus.