

# schoolRISCV + ASCII

[https://github.com/DigitalDesignSchool/ce2020labs/tree/master/day\\_9](https://github.com/DigitalDesignSchool/ce2020labs/tree/master/day_9)

Dmitry Smekhov, 2022

# Цели лабораторной работы

Подключить schoolRISCV к VGA с использованием алфавитно-цифрового дисплея

Используется интерфейс из лабораторной работы **day\_4**

- Изменить контроллер для работы в режиме 1024x768
- Создать компонент видеопамати
- Создать программу пример
- Проверить работу на плате
- Проверить работу на модели

# Два способа — графический и алфавитно-цифровой

- графический — видеопамять хранит значение для каждого пикселя
- алфавитно-цифровой — видеопамять хранит коды СИМВОЛОВ

Главное достоинство алфавитно-цифрового режима это низкие требования к размеру видеопамати и количеству операций с видеопаматью

# Видеопамять для schoolRISCKV

## Компонент **video\_memory**

- Число слов 4096
- Разрядность 8 бит
- Режим заполнения видеопамати начальным значением

## Компонент **text\_rom**

- Текстуры 8x8 для символов ASCII, некоторые символы пропущены

Компонент text\_rom взят из проекта на GitHub:

<https://github.com/PacktPublishing/Learn-FPGA-Programming/tree/master/CH1>

Это проект для очень интересной книги:

**Frank Bruno «FPGA Programming for Beginners»**

ISBN 978-1-78980-541-3

[www.packt.com](http://www.packt.com)

# Режим отображения

Режим отображения:

- SVGA: 1024x768 @60Гц
- Память текстур: 8x8
- Символ: 16x16 (удваивается по вертикали и горизонтали)
- Дисплей: 64x48 символов
- Частота процессора и видеопамяти: 65 МГц

# Шаги лабораторной работы

Step 1 — отображение символа «2»

Step 2 — заполнение видеопамати начальным значением

Step 3 — Вывод «**Hello World!**»

Step 4 — Моделирование проекта

Домашнее задание — добавить запятую в строчку «**Hello,World!**»