**Введение (Слайд 2-3)**

Этот мини-практикум представляет собой экспресс-курс по проектированию цифровой логики и базовой микроархитектуры центрального процессора (ЦП). Используя интегрированную онлайн среду разработки (IDE) Makerchip, вы можете реализовать любые компоненты цифровой системы: от логических вентилей до простого, но полноценного ядра процессора RISC-V. Этот практикум демонстрирует те широчайшие возможности, которыми может пользовать разработчик, используя свободно распространяемые онлайн-инструменты для разработки с открытым исходным кодом. Вы получите фундаментальные навыки для карьеры в области логического проектирования и сможете занять передовые позиции, научившись использовать новое расширение языка Transaction-Level Verilog (TL-Verilog) (даже если вы еще не знаете Verilog).

В этой лекции вы познакомитесь с курсом и его учебной платформой. Вы узнаете:

* какие ресурсы доступны для прохождения этого курса;
* способы взаимодействия с учебной платформой Makerchip.com;
* структуру курса.

**Makerchip IDE и ресурсы**

**Ресурсы курса (Слайд 4)**

Перед началом работы откройте репозиторий GitHub (QR-код на слайде), содержащий внешние ресурсы для этого курса, и прочитайте раздел «Welcome». Там описаны все вносимые изменения в курс. Вам следует добавить эту страницу в закладки или держать ее открытой на протяжении всего курса.

**Makerchip IDE**

Начните работу на сайте makerchip.com (QR-код на слайде). Перейдите на сайт и откройте интегрированную среду разработки, нажав на кнопку “Launch Makerchip IDE”.

Makerchip – это развивающаяся платформа, и версии программы могут отличаться, поэтому возможно, что для вас она будет выглядеть иначе.

Откройте пример. В выпадающем списке «LEARN» найдите различные примеры, а также ряд учебных пособий, которые можно просмотреть в любое время, чтобы закрепить теорию, с которой у вас могли возникнуть трудности, или чтобы освоить материалы, выходящие за рамки того, что будет изучено в этом курсе.

**(Слайд 5)** Откройте пример схемы с делением чисел типа Long. Для этого перейдите в «LEARN», выберите вкладку «Examples». В открывшейся вкладке слева находится пункт «Long Division». Нажмите на кнопку ниже.

Справа в открывшемся окне появится пример шестнадцатеричного деления чисел типа Long.

**(Слайд 6)** Продолжим знакомиться с возможностями Makerchip. Выберите вкладку «EDITOR». Теперь схема деления находится в панели редактора, где можно редактировать модель схемы. Она реализована на транзакционном уровне Verilog; компиляция произошла автоматически, когда был открыт пример.

Чтобы скомпилировать проект, нужно перейти в правый верхний угол редактора, развернуть меню опций и выбрать «Compile/Sim». Альтернативный вариант – запустить компиляцию с помощью комбинации клавиш «Ctrl+Enter».

Процесс компиляции и моделирования, отображаются соответствующими значками в названиях вкладок, как это показано на рисунке ниже.

Во вкладке «LOG» отображается журнал действий. Важно всегда проверять журналы при каждой компиляции и исправлять любые ошибки и предупреждения. Многие ошибки не являются критическими, и проект все равно может скомпилироваться и пройти этап моделирования, но отлаживать ошибки из журнала гораздо проще, чем отлаживать проект во время его моделирования.

**(Слайд 7)** Во вкладке «WAVEFORM» отображается временная диаграмма всех сигналов в системе в течение всего времени симуляции схемы.

**(Слайд 8)** Все представления связаны между собой, поэтому на временной диаграмме можно выбрать какой-нибудь сигнал путем нажатия на него. Выбранный сигнал будет отображаться желтой подсветкой на диаграмме, а также будет выделен цветом на схеме.

Выбранный сигнал будет выделен и на вкладе «NAV-TLV» (рисунок **Ошибка! Источник ссылки не найден.**). Эта вкладка показывает код по аналогии с редактором, но в виде представления кода в интерпретации инструментов компилятора. Для отладки проектов использую именно эту версию кода. В этом коде выделяются ошибки и предупреждения, соответствующие некоторым ошибкам, которые выводятся в журнале компиляции (LOG).

Общий процесс отладки выглядит следующим образом: с помощью временной диаграммы можно найти случаи некорректного поведения схемы при моделировании, и затем их устранить, проследив сигнал на логической схеме и в коде.

Еще есть вкладка «VIZ», которая упрощает отладку больших проектов. О ней, как и о некоторых других элементах IDE, будет рассказано позже.