

Логический анализатор SignalTap

Логический анализатор

Логический анализатор — электронный прибор, который может записывать и отображать последовательности цифровых сигналов



SignalTap

SignalTap — встраиваемый в Cyclone IV FPGA логический анализатор, который позволяет в реальном времени отслеживать сигналы внутри FPGA

- Имеет доступ к любому сигналу в схеме
- Синтезируется вместе с отлаживаемой схемой
- Использует встроенную в FPGA память в качестве буфера
- Передает данные на ПК через JTAG
- Находится в Tools > Signal Tap Logic Analyzer

SignalTap

- JTAG Chain Configuration
- Signal Configuration
- Data/Setup
- Instance Manager

The screenshot displays the Signal Tap Logic Analyzer interface with several key components highlighted:

- Instance Manager:** A table showing the configuration for the 'auto_sigtap_0' instance. The table has columns for Instance, Status, Enabled, LEs, Memory, Small, Medium, and Large. The instance is currently 'Not running'.
- JTAG Chain Configuration:** A dialog box for configuring the JTAG chain, including fields for Hardware, Device, and SOF Manager.
- Signal Configuration:** A dialog box for configuring the signal tap, including settings for Clock, Data (Sample depth, Segmented, Nodes Allocated, Pipeline Factor), Storage qualifier, and Input port.
- Signal Tap Configuration Table:** A table for configuring the signal tap nodes. It has columns for Type, Alias, Name, Lock mode, Data Enable, Trigger Enable, and Trigger Conditions. The table shows one node with 'Basic ANE' as the trigger condition.

The main window shows a Hierarchy Display with 'auto_sigtap_0' listed, and a Data Log window at the bottom.

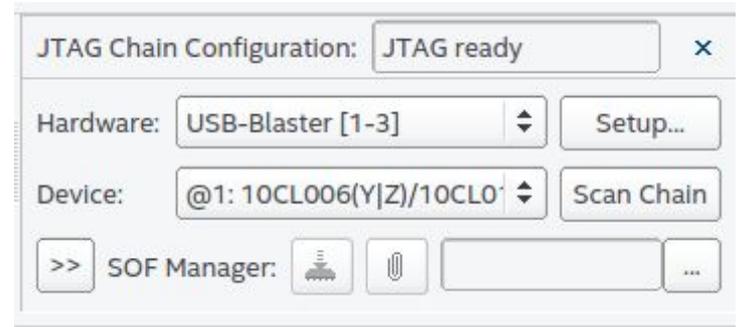
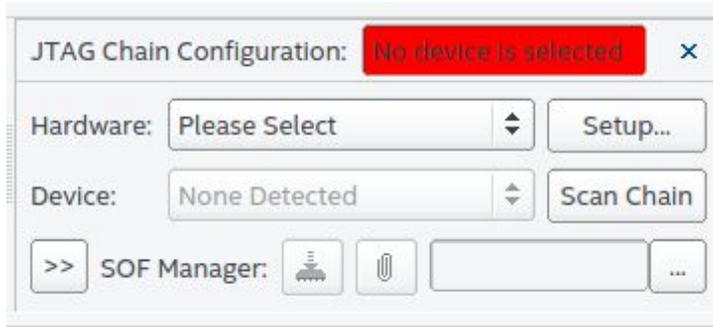
Настройка и использование SignalTap

1. Выбрать JTAG-программатор
2. Добавить желаемые сигналы
3. Выбрать тактовый сигнал
4. Определить количество семплов
5. Запустить компиляцию
6. Загрузить прошивку
7. Запустить анализ сигналов
8. Исследовать временные диаграммы сигналов

Попробуем на примере проекта [examples/2019/07_uart](#)

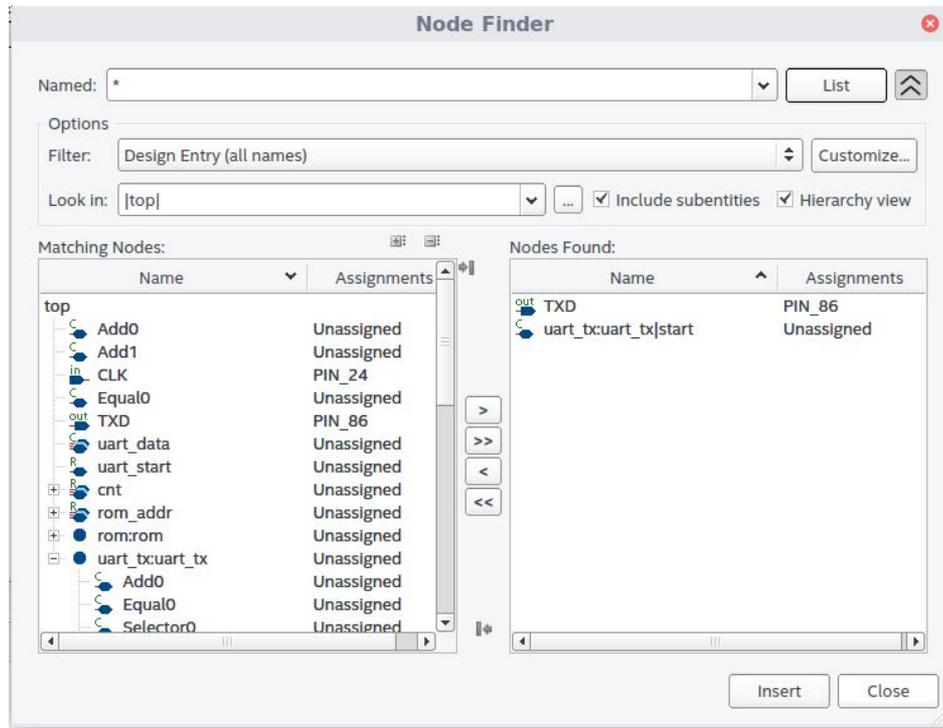
1) JTAG Chain Configuration

- JTAG-программатор выбирается точно так же, как и перед началом обычной прошивки
- Когда JTAG будет настроен, появится надпись “JTAG ready”



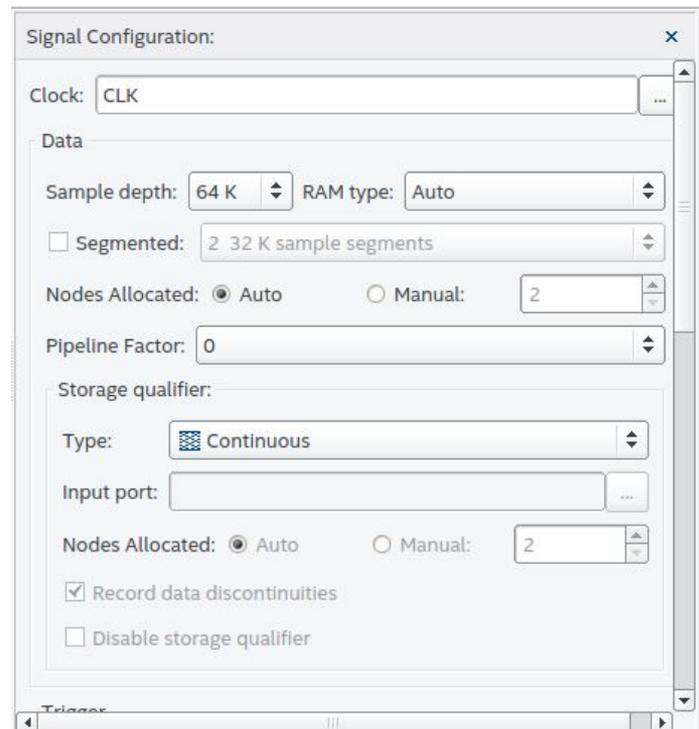
2) Setup

- При двойном нажатии на поле **Setup** открывается окно выбора сигналов
- Можно ввести имя желаемого сигнала или найти его в списке
- После завершения выбора нажать **Insert**
- Сейчас выбраны сигналы **start** из модуля **uart_tx** и выходной пин **TXD**



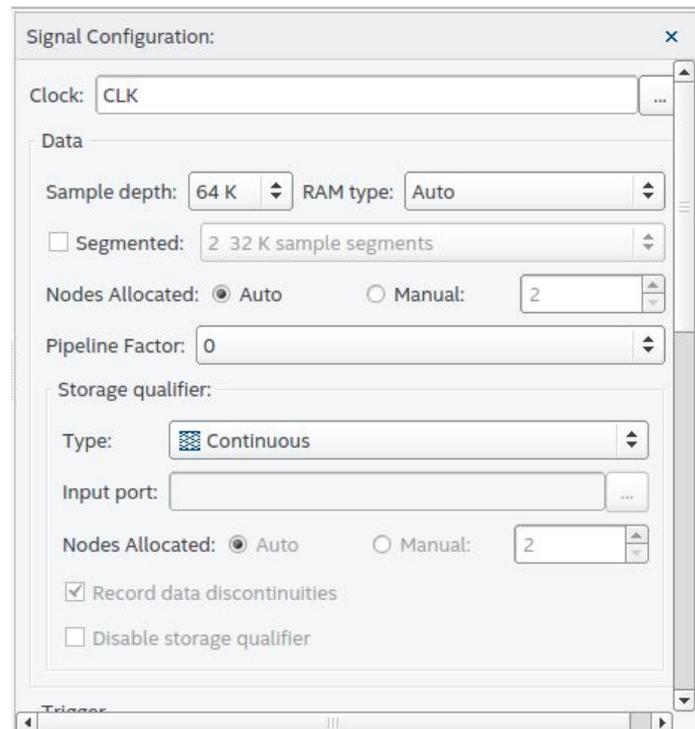
3) Signal Configuration

- В поле Clock нужно выбрать тактовый сигнал, по которому значения исследуемых сигналов будут сохраняться в буфер
- Сейчас выбран сигнал CLK



4) Signal Configuration

- В поле Sample depth нужно выбрать размер буфера — сколько раз будут сохраняться исследуемые сигналы
- В данном случае будет 65536 семплов

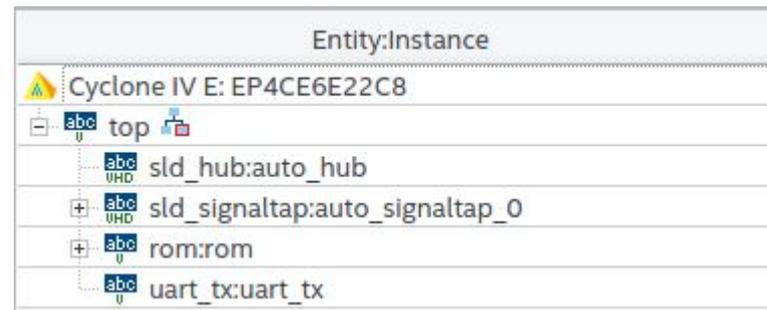
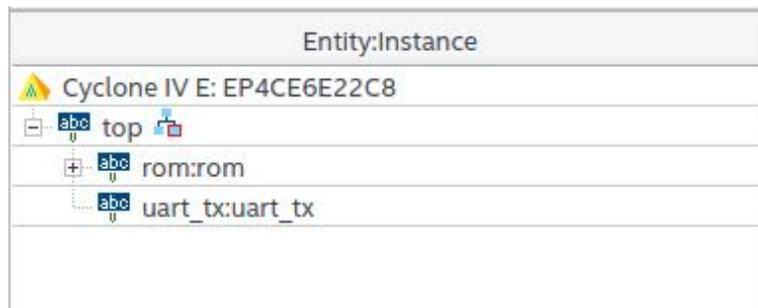


5) Start Compilation

- Сверху есть кнопка Start Compilation

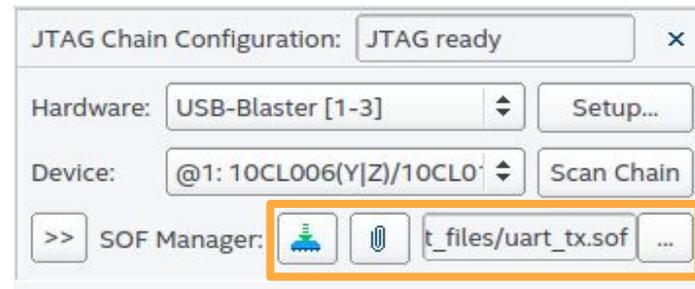


- Теперь в отлаживаемую схему добавится модуль SignalTap

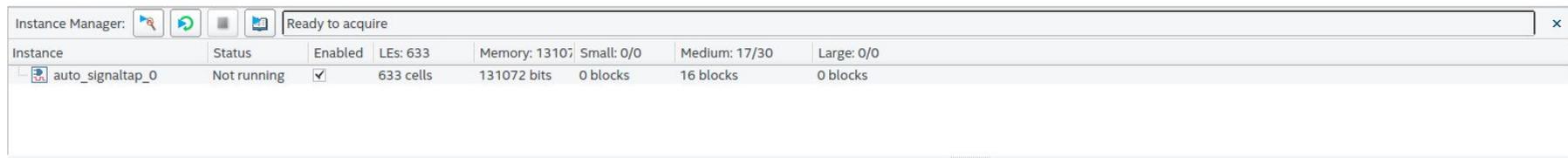


6) JTAG Chain Configuration

- Как и во время обычной прошивки нужно выбрать файл и загрузить его в FPGA



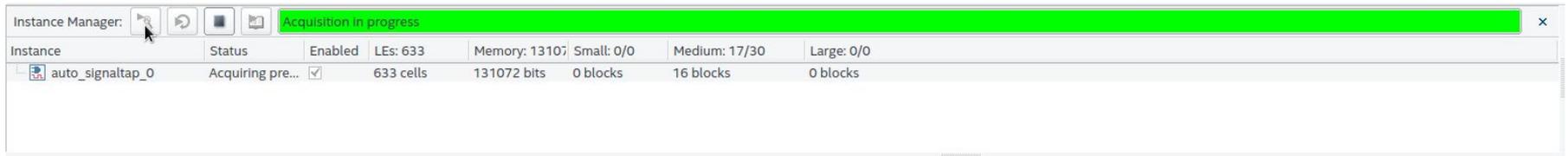
- После этого Instance Manager сообщит о готовности выгружать сигналы на ПК (надпись “Ready to acquire”) и отобразит экземпляры SignalTap (в данном случае он один — `auto_sigtap_0`)



Instance	Status	Enabled	LEs: 633	Memory: 13107	Small: 0/0	Medium: 17/30	Large: 0/0
auto_sigtap_0	Not running	<input checked="" type="checkbox"/>	633 cells	131072 bits	0 blocks	16 blocks	0 blocks

7) Instance Manager

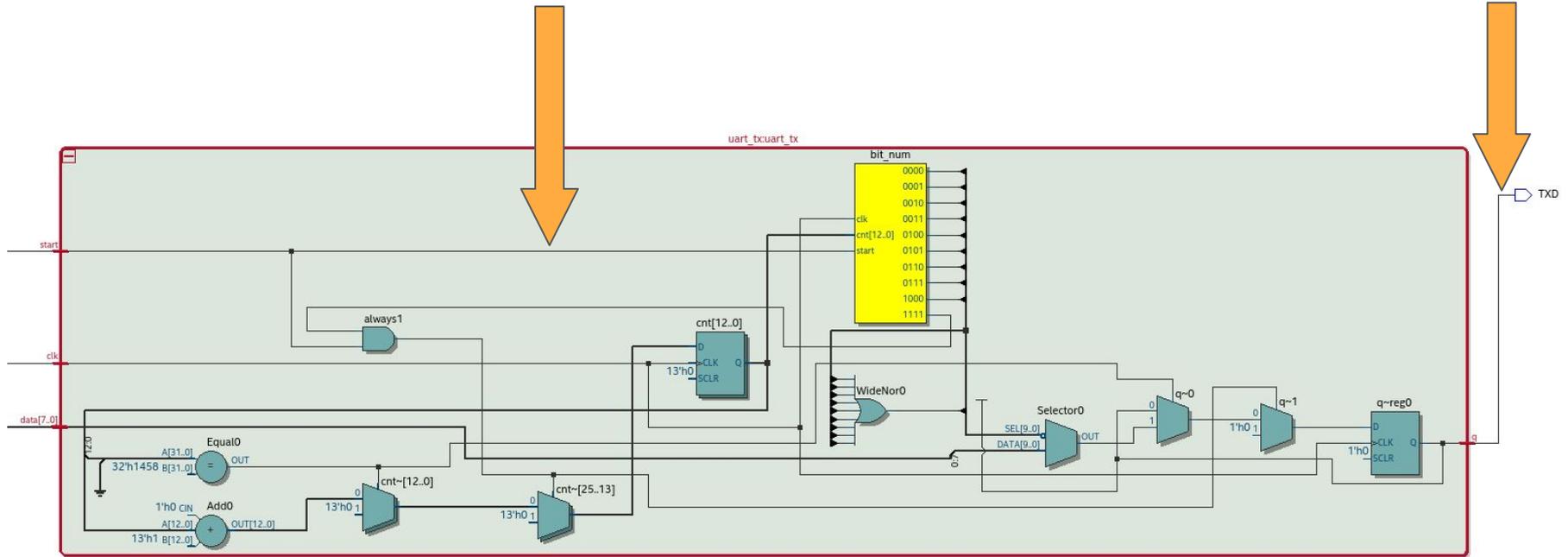
- При нажатии Run Analysis начнется сохранение значений сигналов в память FPGA, после чего они будут отправлены на ПК по JTAG



Cxema

uart_tx | start

TXD



Особенности SignalTap

- Исследование любых сигналов загруженной в FPGA схемы на компьютере без необходимости выводить их на пины
- Сохранение сигналов по определенным пользователем условиям
- Изменение настроек SignalTap требует частичной или полной перекомпиляции проекта и перепрошивки FPGA
- Ограничение на количество семплов, связанное с количеством доступной памяти в FPGA

GitHub

github.com/viktor-prutyanyanov/drec-fpga-intro