

Конфигурирование аппаратного проекта

Вы можете сконфигурировать плагин Nios II для работы в логическом анализаторе SignalTap II. Чтобы сделать это, выполните следующие пункты в программе Quartus II:

Для проекта, загруженного со страницы примера проекта Nios II Ethernet Standard, инкрементная компиляция всегда выключена.

1. В меню Processing выберите **Start** и кликните **Start Analysis & Elaboration**. Эта операция займёт некоторое время.
2. В меню Tools кликните на **SignalTap II Logic Analyzer**.
3. Правым кликом в списке узлов SignalTap II выберите **Add Nodes with Plug-In** и кликните **Nios II**. Появляется диалог **Select Hierarchy Level**.
4. В диалоге Select Hierarchy Level выберите **eth_std_main_system: eth_std_main_system_inst|eth_std_main_system_cpu:cpu**.
5. Кликните **OK**. Появится диалог **Plug-In Options**.
6. В диалоге **Plug-In Options** кликните на кнопку обзора рядом с полем **Setting**.
7. В диалоге **Select File** найдите **.elf** файл, который вы скомпилировали в командной среде Nios II. Этот **.elf** файл должен находиться в директории **software/app/count_binary** и называться **count_binary.elf**. Выберите этот файл.
8. Кликните **Open**. Диалог **Select File** закроется.
9. Кликните **OK**. Диалог **Plug-In Options** закроется.
10. В столбце **Trigger Conditions** плагина Nios II введите адрес, соответствующий операции записи в PIO светодиодов. В нашем примере, адрес - 0x1000038с, для состояния триггера из рис. 1 на стр. 13.




-
11. Добавьте сигнал данных светодиодов PIO в список узлов, выполнив следующие пункты:
 - a. Правым кликом в панели списка узлов кликните **Add Nodes**.
 - b. Проконтролируйте, чтобы поле **Look in** было установлено в `[top_level]eth_std_main_system:eth_std_main_system_inst` и поле **Named** содержало `*led*`.
 - c. Проконтролируйте, что поле **Filter** было установлено в **SignalTap II: pre-synthesis**.
 - d. Кликните **List**.
 - e. На панели **Nodes Found** дважды кликните на узел `led_pio_external_connection_export`. Узел будет добавлен на панель **Selected Nodes**.
 - f. Кликните **OK**.
 12. На панели Signal Configuration кликните на кнопку обзора рядом с текстовым полем **Clock**, чтобы появился диалог **Node Finder**.
 13. В диалоге **Node Finder** выполните следующие пункты:
 - a. Кликните на кнопку обзора рядом с текстовым полем **Look in**. Появится диалог **Select Hierarchy Level**.
 - b. В диалоге **Select Hierarchy Level** раскройте список иерархии под `[top_level]eth_std_main_system:eth_std_main_system_inst`.
 - c. В раскрывшемся списке иерархии кликните на элемент `eth_std_main_system_cpu:cpu`.
 - d. Кликните **OK** для закрытия списка.
 - e. Проследите, чтобы поле **Filter** было установлено в **SignalTap II: pre-synthesis**.
 - f. В поиске узлов кликните на **List** для отображения списка всех узлов.
 - g. Дважды кликните на узел `clk` для добавления его в поле **Selected Nodes**.
 - h. Кликните **OK** для закрытия диалога **Node Finder**.
 14. На панели Signal Configuration установите поле **Sample Depth** в 256.
 15. В окне Quartus II в меню Processing кликните **Start Compilation** для сборки аппаратного проекта. Если вас спросят о сохранении элемента SignalTap II и разрешении логического анализатора SignalTap II в этом проекте, кликните **Yes**.

Вы должны сохранить **.stp** файл в директории вашего проекта.

Компиляция аппаратного проекта Quartus II может занять некоторое время.

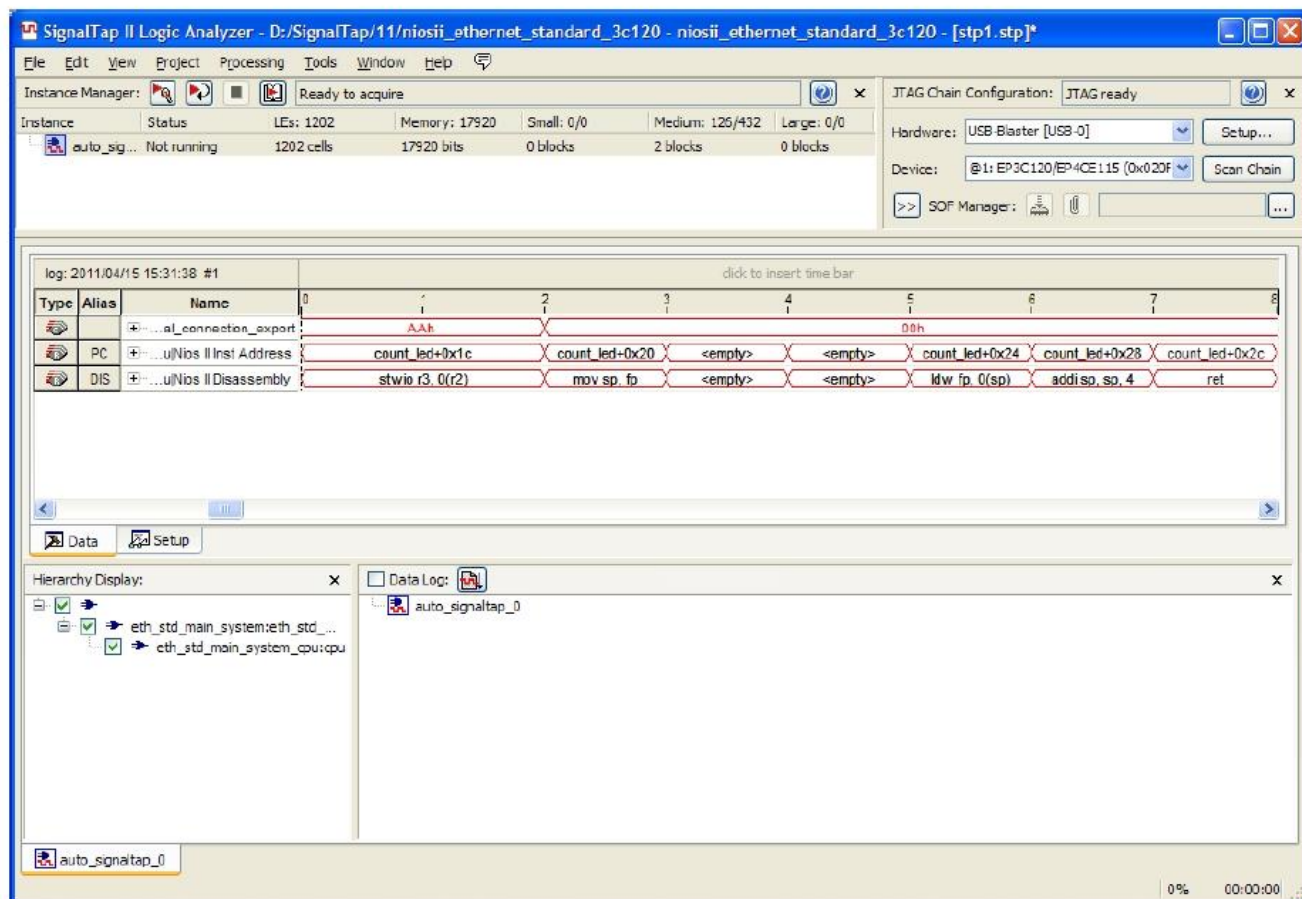
Запуск сессии захвата отведённых данных

Вы можете использовать сгенерированную систему для запуска сессии захвата данных с помощью плагина Nios II, выполнив следующие пункты:

1. В окне SignalTap II на панели JTAG Chain Configuration выполните следующие пункты:
 - a. В меню Hardware выберите подключенный к вашей плате разработчика Nios II кабель программирования.
 - b. В поле **SOF Manager** кликните на кнопку обзор.
 - c. В диалоге **Select Programming File** выберите директорию вашего проекта и **.sof** файл.
 - d. Кликните **Open**. Снова будет доступной кнопка **Program Device**.
 - e. Кликните на кнопку **Program Device** для загрузки **.sof** файла в FPGA .
2. В окне SignalTap II на панели Instance Manager кликните на кнопку **Run Analysis**  для запуска сессии захвата логического анализатора. Анализ может быть остановлен с помощью **Status** на панели Instance Manager, установленным в **Waiting for trigger**.
3. В Nios II SBT на Eclipse выберите проект **count_binary** в списке проектов.
4. В Nios II SBT на Eclipse в меню Run кликните **Debug As**, затем кликните **2 Nios II Hardware**.
5. В диалоге **Confirm Perspective Switch** кликните **Yes**.
6. В Nios II SBT на Eclipse на панели Debug кликните на кнопку **Resume** для запуска исполнения процессора Nios II .

7. Проверьте окно SignalTap II на предмет окончания сессии захвата данных, поскольку получены состояния триггеров плагина Nios II. Ваше окно SignalTap II может иметь следующий вид (рис. 2).

Figure 2. SignalTap II Post Capture Window



Анализ собранных данных

Теперь вы можете анализировать собранные данные и сравнивать их с **objdump** из Nios II **.elf** файла. Собранные отсчёты в плагине Nios II ведут себя в соответствии с программой на ассемблере из файла **count_binary.objdump**.

На рис. 2 отображены следующие результаты:

- Отсчёт 0th захватывается плагином Nios II с созданным состоянием триггера, которое вы задали, захвачена инструкция **stwio r3,0(r2)**.

Заключение

Поскольку FPGA проекты продолжают наращивать размер и сложность, увеличивая количество встроенных процессоров, периферии, шин и мостов, разработчикам необходим комплексный инструмент отладки для уменьшения времени разработки системы. Плагин Nios II для встроенного логического анализатора SignalTap II позволяет захватывать данные при исполнении программы в процессоре Nios II. Поскольку плагин Nios II работает совместно с логическим анализатором SignalTap II, то инструкции, отведённые от процессора Nios II, захватываются другими аппаратными элементами. Отображение инструкций процессора позволяет вам эффективно находить проблемы в проекте системы.