

## ➤ ПРОГРАММИРОВАНИЕ И ОТЛАДКА ЛОГИКИ ПЛИС НА СТЕНДЕ NANOBOARD

Для программирования коммутации ПЛИС и отладки логики FPGA-проекта компания Altium Ltd. разработала семейство отладочных стендов NanoBoard, использование которых превращает программно-аппаратный комплекс Altium Designer в "инновационную станцию" (Altium Innovation Station), на которой может быть выполнен полный цикл разработки проекта ПЛИС, включая программирование тиража микросхем для установки их в печатные платы, подлежащие поставке заказчику.

### Ресурсы NanoBoard

На рынок последовательно поступали стенды NanoBoard NB1, NanoBoard NB2DSK01 и NanoBoard 3000<sup>1</sup> (рис. 1). По сути, каждая из моделей представляет собой специализированный ком-

пьютер с развитой периферией и средствами подключения отлаживаемой микросхемы к аппаратным средствам стенда.



Рис. 1. Отладочный стенд NanoBoard 3000

"Сердцем" отладочного стенда является контроллер, осуществляющий связь исполняемого FPGA-проекта Altium Designer с "физической" микросхемой ПЛИС, установленной на стенде для программирования коммутации внутренних связей и отладки логики проекта. Протокол обмена данными Altium Designer с отладочным стендом NanoBoard, разработанный компанией Altium Ltd., называется NanoTalk, а контроллер стенда — NanoTalk Controller.

В моделях NanoBoard NB1 и NanoBoard NB2DSK01 отлаживаемая ПЛИС устанавливается на сменных "дочерних" платах (Daughter Board), а на самой материнской плате стенда располагаются только контроллер, управляющий процессом отладки и программирования коммутации ПЛИС, органы индика-

<sup>1</sup> Подробная информация размещена на сайте [www.altium.com/nanoboard](http://www.altium.com/nanoboard).

ции, средства, имитирующие связь ПЛИС с ее внешним окружением в проектируемом функциональном узле и средства последовательного наращивания, обеспечивающие возможность использования сразу нескольких стенов NanoBoard в одной конфигурации.

В моделях семейства NanoBoard 3000 на материнской плате стенда смонтирована микросхема ПЛИС одного из трех производителей — компаний Xilinx (NanoBoard 3000XN), Altera (NanoBoard 3000AL) или Lattice (NanoBoard 3000LC), которая названа "пользовательской" ПЛИС (User FPGA) и может применяться в качестве одной из реализаций FPGA-проекта.

Далее мы постараемся рассказать о функциональных возможностях и технологии использования отладочного стенда NanoBoard 3000AL с предустановленной "пользовательской" ПЛИС семейства Cyclone III, EP3C40F780C8N компании Altera. При реализации проекта на ПЛИС другого типа предусмотрена возможность установить ее на материнской плате на сменной "периферийной" плате (Peripheral Board).

Связь стенов NanoBoard с ПК осуществляется через стандартный USB-порт по мультиплексированному каналу JTAG-интерфейса. По этому каналу в ПК и из ПК передается информация, которая служит для управления работой контроллера стенда (NanoTalk Channel), программирования внутренней коммутации пользовательской ПЛИС (Hard Devices Channel) и загрузки в пользовательскую ПЛИС программного приложения при проектировании микроконтроллеров или микро-ЭВМ (Soft Devices Channel). При подключении стенда NanoBoard к ПК с работающей программой Altium Designer автоматически активируется драйвер, опознающий наличие и идентифицирующий версию NanoBoard и периферийной платы (если она подключена) и устанавливающий связь активного FPGA-проекта с ресурсами NanoBoard.

Для обнаружения и идентификации версии NanoBoard и периферийной платы на них устанавливаются ПЗУ (1-Wire Memory), в которые занесен 40-байтовый идентификационный код.

### Взаимодействие FPGA-проекта с ресурсами NanoBoard

Доступ активного FPGA-проекта к ресурсам NanoBoard, в том числе — к пользовательской ПЛИС, осуществляется из оболочки *Devices View* в режиме *Live*.

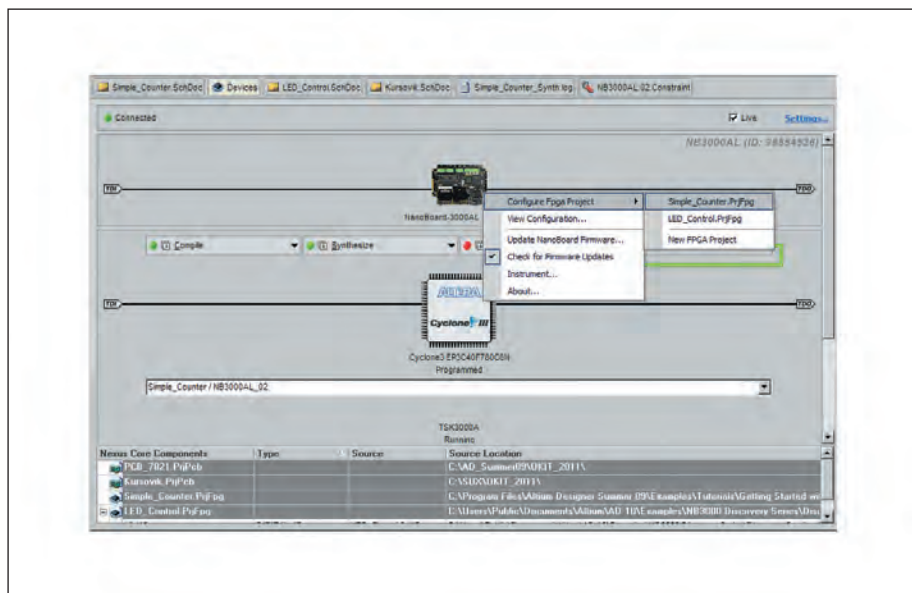


Рис. 2. Панель-оболочка *Devices View*

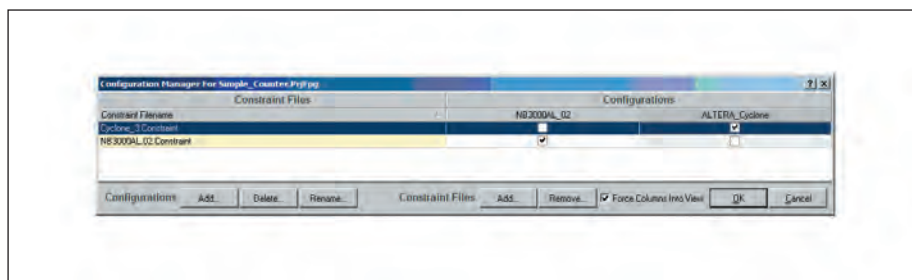


Рис. 3. Диалог настройки конфигурации FPGA-проекта

Для обеспечения доступа должна быть сформирована конфигурация проекта. Это, как уже отмечалось ранее, именованный список файлов привязки (*Constraint files*). Файлы привязки, описывающие ресурсы NanoBoard, как и конфигурация, могут быть сформированы вручную, однако Altium Designer предоставляет возможность упростить данную трудоемкую работу путем автоматического формирования конфигурации. Для этого нужно щелчком правой клавишей мыши на значке *NanoBoard* в панели *Devices View* активировать контекстное меню и указать в нем команду *Configure FPGA Project* (рис. 2). В открывшемся подменю следует выбрать один из списка загруженных проектов или указать команду *New FPGA Project* (Создать новый FPGA-проект).

При выборе имени существующего проекта программа находит файлы привязок (*Constraint Files*), загруженные при инсталляции Altium Designer в директорию ... \Library\FPGA\ и автоматически подключает их к проекту в соот-

ветствии с обнаруженным системой идентификационным кодом NanoBoard. Так, при обнаружении системой стенда NanoBoard 3000 с идентификационным кодом <ClassID> = NB3000AL.02 программа извлекает из директории файл NB3000AL.02.Constraint. При обнаружении отладочной периферийной платы PB30.01 программа дополнительно подключает к FPGA-проекту файл привязок PB30.01.Constraint.

Командой главного меню *Project>Configuration Manager* или *View Configuration* того же контекстного меню открывается диалог просмотра и редактирования конфигурации проекта (рис. 3).

При использовании отладочной периферийной платы программа автоматически формирует и включает в конфигурацию файл привязок, содержащий карту подключения периферийной платы к материнской плате NanoBoard 3000. Имя файла — NB3000AL\_02\_BoardMapping.Constraint. Этот файл сохраняется в той же директории, что и файл проекта \*.PrjFpg.



## Порт-компоненты

Для связи цепей схемного документа РСВ-проекта с цепями логического ядра в схеме FPGA-проекта к этим цепям подключаются одноименные метки *Port*. Затем в файлах привязок (*Constraint*) назначается подключение этих цепей к контактам выбранной ПЛИС. При работе с отладочным стендом NanoBoard 3000 подключение внешних выводов «пользовательской» ПЛИС выполняется печатным монтажом на материнской плате, поэтому нет необходимости подключать к внешним связям схемы FPGA-проекта стандартные порты и назначать их привязку к выводам «пользовательской» ПЛИС. Вместо этого для имитации сигналов внешнего окружения ПЛИС в Altium Designer сформированы библиотеки специальных интерфейсных компонентов, которые должны быть включены в схемный документ FPGA-проекта. Эти компоненты, называемые в лексике Altium Designer порт-компонентами, размещены (для NanoBoard 3000) в библиотеке ...\\Library\\FPGA\\FPGA NB3000 Port-Plugin.IntLib.

Такие интерфейсные компоненты обеспечивают автоматическое подключение ресурсов NanoBoard к входным/выходным контактам пользовательской микросхемы ПЛИС и по присвоенному им по умолчанию параметру *PortComponent True* распознаются программой как внешние по отношению к FPGA-проекту.

Компоненты *Nexus\_JTAG\_Connector* и *Nexus\_JTAG\_Port* обеспечивают доступ Altium Designer к встроенным в «пользовательскую» ПЛИС программным приложениям и виртуальным инструментам отладки. Сигналы JTAG-интерфейса порождаются контроллером стенда и подводятся к четырем предопределенным контактам пользовательской ПЛИС.

Порт-компоненты автоматически конвертируются в стандартные метки *Port* при выполнении синтеза проекта, обеспечивая связность логического ядра проекта с цепями внешнего окружения ПЛИС при объединении FPGA- и РСВ-проектов.

## Программирование FPGA-проекта на «пользовательской» ПЛИС стенда NanoBoard 3000

Рассмотрим стадии программирования ПЛИС на примере проекта генератора двоичной псевдослучайной последовательности (ПСП) длиной 8 бит (рис. 4).

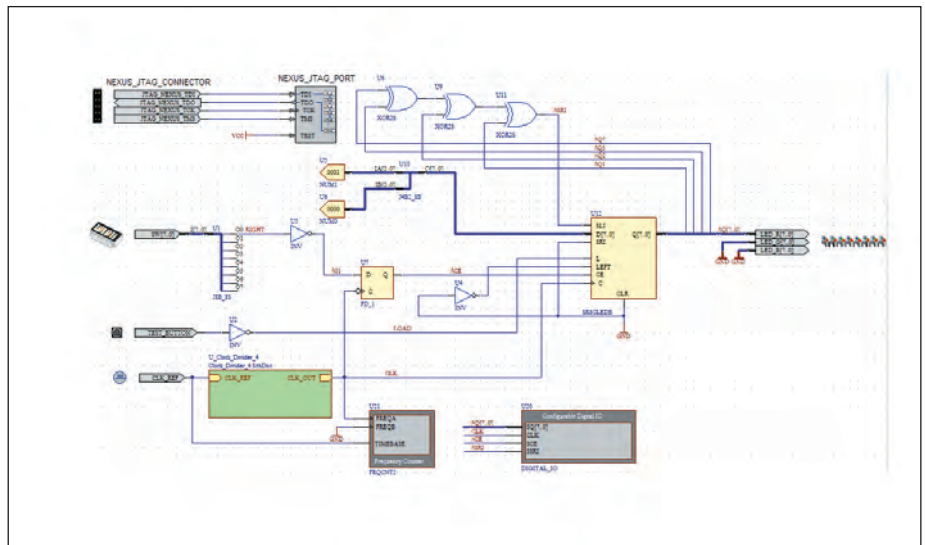


Рис. 4. Функциональная схема генератора ПСП с порт-компонентами и виртуальными инструментами отладки

Функциональная схема генератора ПСП построена на базе логических прототипов из библиотеки *FPGA Generic.IntLib*.

Для проверки правильности функционирования и демонстрации работы генератора ПСП на стенде NanoBoard 3000 ко входам логики и выходам регистра подключены следующие порт-компоненты из библиотеки *FPGA NB3000 Port-Plugin.IntLib*:

- **CLK\_REF** — интерфейс с тактовым генератором стенда NanoBoard с частотой 20 МГц;
- **TEST\_BUTTON** — кнопка общего сброса *TEST/RESET*;
- **DIPSWITCH** — DIP-блок из восьми переключателей (SW8);
- **LEDS\_RGB** — блок 3x8 «пользовательских» светодиодов красного, зеленого и синего цветов (*RGB USER LEDS*).

Чтобы мелькание светодиодов, подключенных к выходам разрядов регистра, воспринималось глазом, в схему введен делитель частоты *U\_Clock\_Divider* с коэффициентом деления  $10^7$  (частота импульсов на выходе *CLK\_OUT* составит 2 Гц). Делитель частоты может быть построен на библиотечных логических прототипах или в виде HDL-файла в конструкциях языка описания аппаратуры VHDL. В обоих случаях он должен быть преобразован командой меню *Design>Create Symbol from Sheet or HDL* в иерархический элемент и включается в дерево документов проекта (рис. 5) как файл с именем *Clock\_Divider\_4.SchDoc* или *Clock\_Divider.vhd*.

После сохранения и компиляции документов проекта следует выполнить программирование ПЛИС в оболочке *Devices View*:

- командой *View>Devices View* открываем оболочку *Devices View*;
- командой контекстного меню *Configure FPGA Project* (рис. 2) формируем конфигурацию проекта (имя конфигурации — *NB3000AL\_02*). В дерево проекта программа включает файл привязок *NB3000AL\_02.Constraint*;

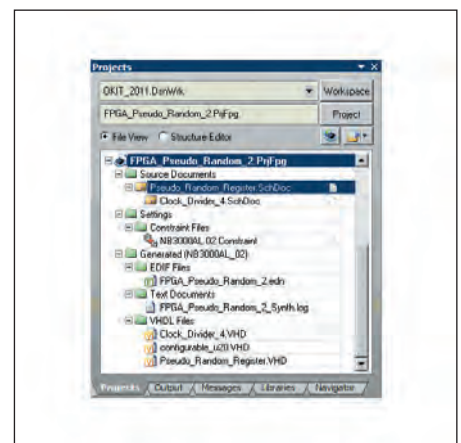


Рис. 5. Дерево документов FPGA-проекта

- щелчком правой клавишей мыши на значке с изображением ПЛИС в поле *Hard Devices Chain* активируем контекстное меню и указываем команду *Reset* — сброс коммутации, ранее нанесенной в ПЛИС;

■ выполняем последовательные стадии обработки проекта:

- *Compile* — компиляция документов проекта;
- *Synthesize* — синтез промежуточного VHDL-описания \*.vhd и исходного файла описания схемы \*.edn для программирования в среде САПР-производителя ПЛИС (Vendor Tools) в обменном формате EDIF. Оба файла, а также файл протокола \*.log включаются программой в дерево документов проекта;
- *Build* — трансляция логической структуры проекта в логические примитивы ПЛИС и формирование файла программирования коммутации внутренней структуры ПЛИС;
- *Program FPGA* — загрузка файла программирования в микросхему ПЛИС на панели стенда NanoBoard 3000 или на съемной отладочной плате PB30.

Процесс выполнения всех фаз обработки проекта сопровождается выводом сообщений в панель *Messages*. Более подробная информация о прохождении всех фаз обработки, включая сообщения САПР *Vendor Tools* (в нашем случае это программа Altera QUARTUS II), отображается в панели *Output*.

По окончании загрузки программы коммутации надпись *Reset* (Сброшено) под значком микросхемы ПЛИС в панели *Devices View* заменяется на *Programmed* (Запрограммировано), а на материнской панели NanoBoard загорается светодиод LED2 PGM (Programmed). Можно запускать генератор ПСП и проверять его работу.

- Кнопкой *Test/Reset* в триггеры регистра загружается двоичная константа 00000001.
- Перемещением движка переключателя SW7 в положение ON (Включено) запускается сдвиг справа налево. Светодиоды RGB\_USER\_LEDS отображают перемещение «единиц» и «нулей» по разрядам регистра.
- Перемещением движка переключателя SW7 в положение OFF (Выключено) работа генератора ПСП приостанавливается.

Программа коммутации логики, занесенная в ПЛИС на стенде NanoBoard 3000, действует до тех пор, пока включено питание и после выключения питания теряется. Поэтому для повторного запуска проекта на стенде NanoBoard

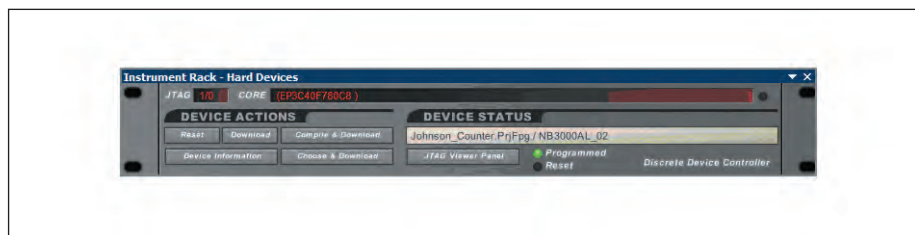


Рис. 6. Виртуальная панель инструментов работы с ПЛИС

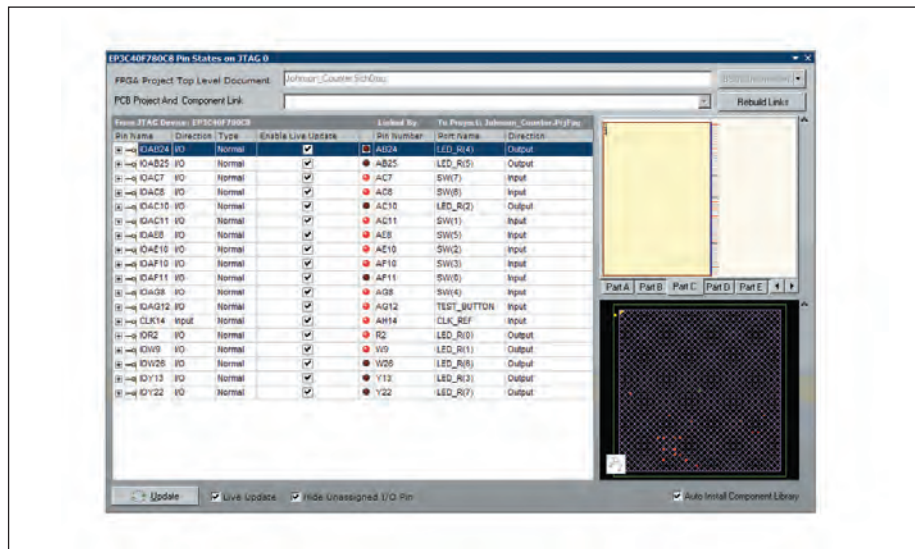


Рис. 7. Панель просмотра внешних выводов ПЛИС

3000 требуется повторная загрузка программы коммутации в оболочке *Devices View*.

### Контроль сигналов на внешних выводах ПЛИС

Функции оболочки *Devices View* позволяют контролировать сигналы на внешних выводах ПЛИС во время ее работы на стенде NanoBoard. Контроль выполняется средствами JTAG-канала, доставляющего эти сигналы с работающей ПЛИС в среду Altium Designer. Для проведения такого контроля следует:

- щелчком правой клавишей мыши на значке ПЛИС в поле *Hard Devices Chain* активировать контекстное меню и указать команду *Instrument*. Открывается панель виртуальных инструментов отладки (рис. 6);
- щелчком левой клавишей мыши на кнопке *JTAG Viewer Panel* активировать панель просмотра внешних выводов ПЛИС (рис. 7);
- установить флажки опций *Hide Unassigned I/O Pin* (Скрыть неиспользуемые входные/выходные выводы) и *Live Update* (Динамически отображать состояние);

В результате в колонке *Pin Number* красными обозначениями динамически отображается состояние активности (уровень логической единицы) используемых выводов ПЛИС. Оно же воспроизводится подсветкой красного цвета на выводах, расположенных в правом поле панели логического символа и посадочного места микросхемы.

### Подключение виртуальных инструментов отладки

Для проверки прохождения логических сигналов по узлам внутренней логики ПЛИС могут использоваться виртуальные инструменты отладки. Логические прототипы инструментов отладки собраны в библиотеке ... \Library\FPGA\FPGA Instruments.IntLib. В терминологии Altium Designer они представляют собой Core-компоненты — результаты выполнения Core-проектов.

- Вводим в логическую схему FPGA-проекта (рис. 4) виртуальные инструменты отладки — счетчик-частотомер FRQCNT2 и модуль ввода/вывода цифровых сигналов DIGITAL\_IO.
- Для поддержки доступа виртуальных инструментов к ресурсам JTAG-канала в схему введем также два ин-

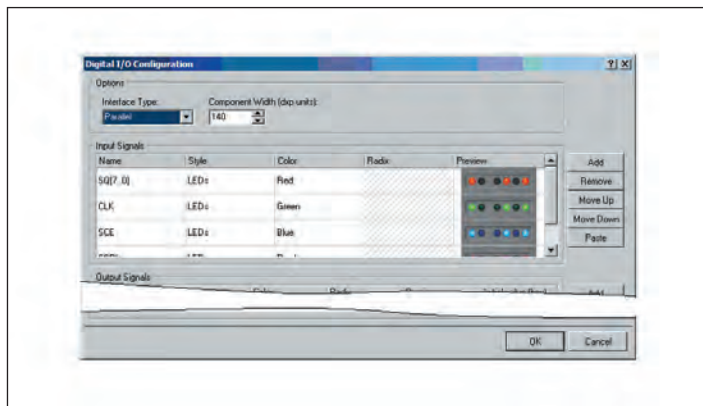


Рис. 8. Диалог настройки конфигурации модуля цифрового ввода/вывода

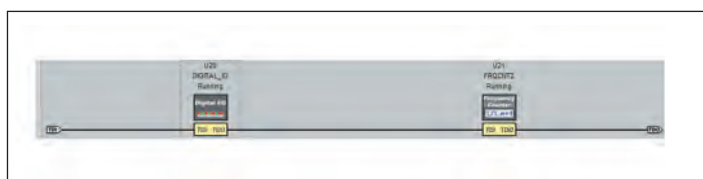


Рис. 9. Цепочка *Soft Devices* после подключения виртуальных инструментов отладки

терфейсных порт-компонента: *NEXUS\_JTAG\_CONNECTOR* – виртуальный соединитель JTAG-канала и *NEXUS\_JTAG\_PORT* – виртуальный порт приема сигналов JTAG-канала.

- Выполним электрические связи входа и выхода виртуального частотомера и портов JTAG-канала, как показано на схеме (рис. 4).
- Щелчком правой клавишей мыши на схемном символе модуля U20 активируем контекстное меню и укажем команду *Configure U20 (DIGITAL IO)*. Открывается диалог настройки конфигурации модуля цифрового ввода/вывода (рис. 8).
- В поле *Input Signals* диалогового окна *Digital IO Configure*, используя кнопку *Add*, составим список сигналов, подлежащих контролю.
- Выполним электрические соединения выводов виртуального модуля U20 с соответствующими узлами схемы. Проводники можно не прочерчивать, а вместо них установить на сигнальные выводы модуля соответствующие метки цепей.
- Сохраним и откомпилируем документы проекта и активируем оболочку *Devices View*.
- Повторим в оболочке *Devices View* стадии обработки проекта. В резуль-

тате после загрузки файла коммутации в ПЛИС в поле *Soft Devices Chain* появится изображение канала виртуальных инструментов *Soft JTAG Channel* (рис. 9).

- Кнопкой *Test/Reset* загрузим в регистр U12 начальную комбинацию 00000001 и движком переключателя SW7 запустим генератор ПСП.
- Двойными щелчками левой клавишей мыши на значках виртуальных инструментов активируем панель *Instrument Rack* (рис. 10).

В панели частотомера отображается число перепадов тактового сигнала CLK на входе C регистра генератора ПСП.

### Логический анализатор

Логический анализатор *LAX* – компонент библиотеки *FPGA Instruments.IntLib* (рис. 11), – как и рассмотренные ранее виртуальные инструменты отладки, представляет собой конфигури-



Рис. 10. Панель виртуальных инструментов отладки. В панели виртуального цифрового модуля световыми индикаторами отображаются логические уровни сигналов на внутренних узлах ПЛИС и выходных контактах. Одновременно в левой части поля *SQ[7..0]* выводится шестнадцатеричное число, показывающее состояние разрядов регистра

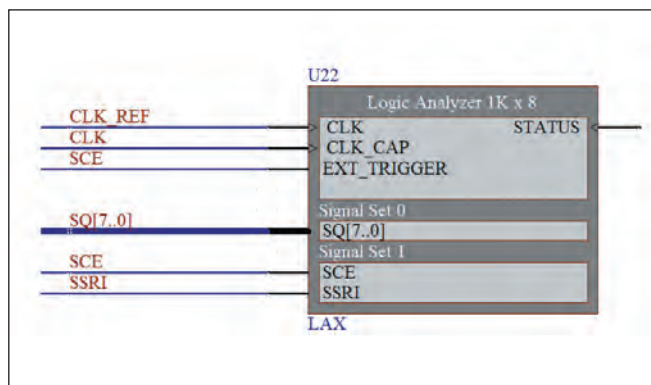


Рис. 11. Схемный символ логического анализатора *LAX*

руемый Core-компонент, размещенный на избыточных ресурсах кристалла ПЛИС, и подключается при программировании коммутации к назначенным в его конфигурации сигнальным узлам логической схемы FPGA-проекта. Анализатор набирает заданное число выборок сигналов логики либо набирает выборки в течение заданного временного интервала, после чего может быть выполнена визуализация набранного массива в виде таблицы или в виде временных диаграмм сигналов<sup>2</sup>.

В верхнем, неизменяемом поле схемного символа расположены четыре сигнальных вывода:

- *CLK* – вход внешнего тактового сигнала;
- *CLK\_CAP* – вход тактового сигнала меньшей, чем CLK, частоты, с которой происходит захват выборок;
- *EXT\_TRIGGER* – вход внешнего управления стартом выполнения вы-

<sup>2</sup> Документ CR0103LAX\_x Lagic Analyzer.pdf Core Reference CR0103 (v2.0) March 17, 2008.



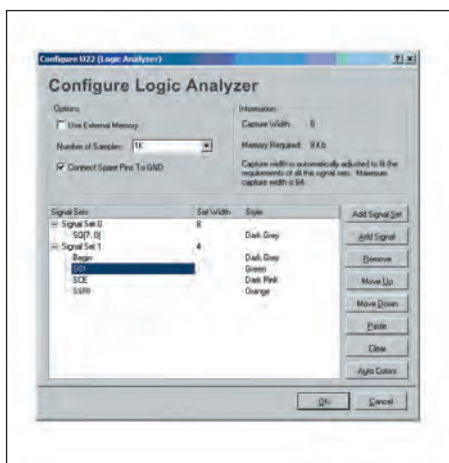


Рис. 12. Настройка конфигурации логического анализатора

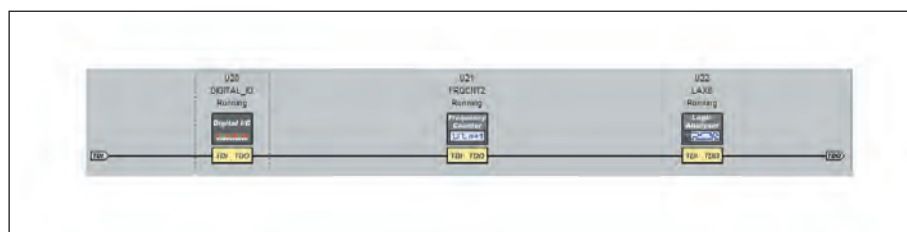


Рис. 13. Цепь виртуальных инструментов отладки ПЛИС



Рис. 14. Панель управления логическим анализатором

борок (при заземлении этого входа выборки отбираются под действием внутреннего управления);

- **STATUS** – выход сигнала статуса: 1 во время отбора выборок и 0 в состоянии покоя.

Нижние поля формируются при настройке конфигурации анализатора. Настройка конфигурации активируется командой *Configure Uxx (LAX)* контекстного меню, вызываемого щелчком правой клавишей мыши на схемном символе анализатора *Uxx*.

В поле *Options* (рис. 12) следует выбрать из выпадающего списка *Number of Samples* число выборок сигнала для накопления в памяти анализатора – *1K*, *2K* или *4K*, если установлен флажок опции *Use External Memory*. Для накопления выборок активируется интерфейс с внешней памятью и разрядностью адресной шины до 20 разрядов.

В центральном поле диалогового окна, используя кнопки *Add Signal Set (Добавить набор сигналов)* и *Add Signal (Добавить сигнал)*, составим список узлов логической схемы (список сигналов), подлежащих контролю.

Сигналы могут быть сгруппированы в наборы (*Signal Sets*). При составлении списка программа автоматически, шагами, кратными 8, увеличивает число входных линий анализатора. Максимально возможное число входных линий – 64. Информация об этом, а также о соответствующем объеме встроенной памяти анализатора отображается в поле *Info* диалогового окна. В одном сеансе измерений программа отбирает выборки сигналов одного набора.

Для нашего примера образуем два набора сигналов:

- в первый набор *Signal Set 0* направим восемь выходных сигнальных линий регистра сдвига *SQ[7..0]*;
- во второй набор *Signal Set 1* направим три сигнала: пуска ПСП (*S0*), разрешения сдвигов (*SCE*) и сигнал обратной связи *SSRI*, загружающий единицу в младший разряд регистра.

Затем следует сохранить документы проекта, войти в оболочку *Devices View* и, как обычно, выполнить программирование ПЛИС. По окончании программирования в цепочке *Soft Devices Chain* появляется значок, обозначающий присутствие в конфигурации логического анализатора (рис. 13). Двойным щелчком левой клавишей мыши на значке анализатора активизируем панель виртуальных инструментов с органами управления логическим анализатором (рис. 14). Управление выполняется кнопками панели:

- **Reset** – начальная установка анализатора;
- **Arm** – привести анализатор в режим ожидания сигнала внешнего управления или запуска по совпадению

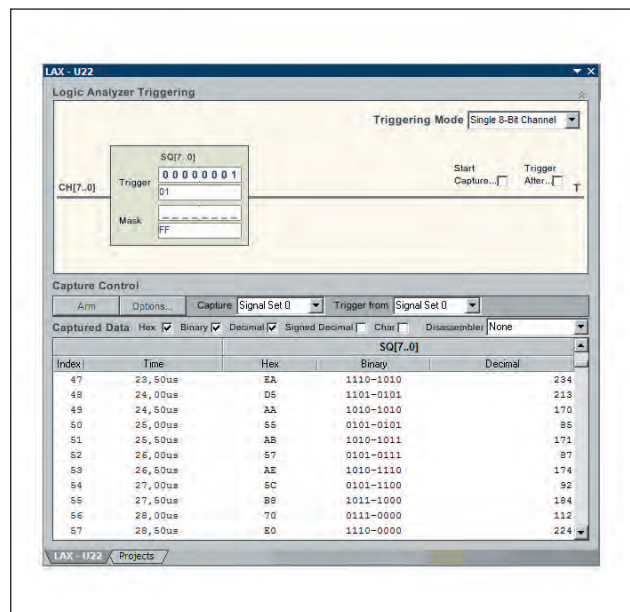


Рис. 15. Плавающая панель логического анализатора

кода на сигнальных линиях с маской пуска в режиме внутреннего управления;

- **Show Waves** – две кнопки, открывающие просмотр временных диаграмм исследуемых сигналов в аналоговой (*Analog*) или дискретной (*Digital*) форме;
- **Options** – щелчком левой клавишей мыши на этой кнопке открывается диалог настройки режима анализа. В нашем примере установим флажок опции *Enable External Trigger (Разрешить внешнее управление набором выборок)*, кроме того, выберем периодичность обновления изображений *Update Display from Core Every 200 ms (Восстанавливать изображение*

ние каждые 200 мс). Укажем также набор сигналов для отображения — *Signal Set 0*.

- **Show Panel (Показать панель)** — щелчок левой клавишей мыши на этой кнопке открывает плавающую панель рабочего пространства *LAX* (рис. 15), содержащую еще ряд настроек логического анализатора и отображающую набранную выборку:

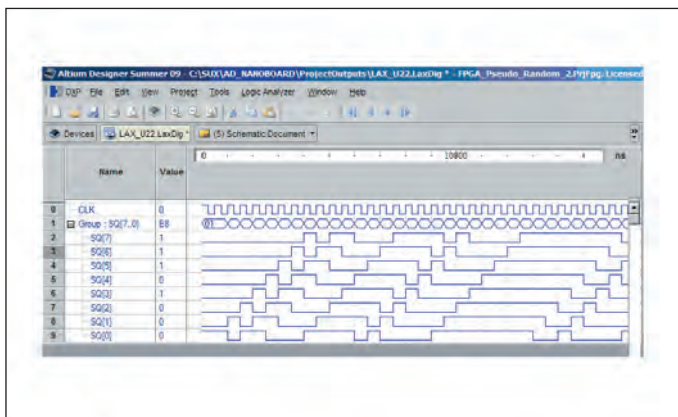


Рис. 16. Главное окно программы с отображением временных диаграмм

- в верхнем поле панели *Logic Analyzer Triggering* находятся функции настройки анализатора в режиме набора выборок под действием внутренней коммутации (*Internal Triggering*);
- в среднем поле *Capture Control* располагаются кнопки, дублирующие кнопку готовности *Arm* и кнопку настроек *Options*, а также поля управления *Capture* — поле назначения набора сигналов для взятия выборок — и *Trigger from* — поле назначения набора сигналов, из кодов которого формируется команда пуска анализатора;
- в нижнем поле *Captured Data* назначается способ отображения данных, попавших в выборку: в виде шестнадцатеричных (*Hex*), двоичных (*Binary*), десятичных (*Decimal*) чисел без знака или со знаком, текстовых символов. После набора выборки отображается весь массив набранных данных.

В нашем примере выбран способ внешнего управления набором выборки, поэтому кнопкой *Arm* анализатор приводится в готовность, а набор выборки на-

чинается по сигналу пуска генератора ПСП от переключателя *SW1* на панели NanoBoard 3000.

Отображение результатов в виде временных диаграмм в аналоговой (*Analog*) или цифровой форме (*Digital*) активируется щелчками левой клавишей мыши на кнопках *Show Waves (Показать волны)*. Программа формирует файлы *LAX\_Uxx.LaxAn* или *LAX\_Uxx.LaxDig* и строит графическое отображение данных в главном окне (рис. 16).

Если позволяют свободные ресурсы кристалла ПЛИС, к проекту может быть подключено несколько логических анализаторов. В этом случае все они включаются в цепочку *Soft Devices Chain* в оболочке *Devices View*, для каждого из них может быть активирована своя панель

управления, и отбор выборок производится анализаторами независимо друг от друга.

В заключение отметим преимущества и недостатки рассмотренной конфигурации инновационной станции, объединяющей программный комплекс Altium Designer с ресурсами отладочного стенда NanoBoard 3000.

К преимуществам следует отнести наличие развитой периферии, реализованной на материнской плате отладочных стендов NanoBoard. Это позволяет разрабатывать и отлаживать сложные иерархические FPGA-проекты, включающие на нижнем уровне иерархии проекты ядер (Core Project), таких, например, как контроллеры интерфейса RS232, RS485 USB, Ethernet и других протоколов, контроллеры клавиатуры, дисплея, АЦП/ЦАП, цифрового ввода-вывода аудиосигнала и др.

Несомненным преимуществом заложенной в Altium Designer технологии является возможность возврата запрограммированной ПЛИС в РСВ-проект благодаря функциям Мастера объединения *FPGA to PCB Project Wizard*, что позволяет выполнять сквозной проект радиоэлектронного функционального

узла на печатной плате, объединяющий в своей иерархии РСВ-проект, подчиненный ему один или несколько FPGA-проектов, вложенные в FPGA Core- и Embedded-проекты.

Вместе с тем следует отметить и ряд недостатков, ограничивающих сложность проектов, которые можно разработать в рассматриваемой среде:

- наличие на материнской плате NanoBoard 3000AL несменяемой «пользовательской» ПЛИС Altera Cyclone III EP3C40F780C8N позволяет разрабатывать и отлаживать FPGA-проекты только на этой микросхеме. От этого недостатка свободен отладочный стенд NanoBoard NB2DSK01;
- «пользовательская» ПЛИС стенда NanoBoard 3000AL имеет 780 контактов, из которых только 535 доступны для программирования пользователем, в то время как в настоящий момент на рынок поступают микросхемы ПЛИС с числом выводов 1000 и более;
- съемная периферийная плата PB30 позволяет разрабатывать проекты на микросхемах других семейств и типов того же производителя — компании Altera, но для программирования пользователю доступны только 50 входных/выходных контактов пользовательской ПЛИС, чего достаточно далеко не для всех возможных применений;
- при обработке проекта в оболочке *Devices View* не полностью используются ресурсы САПР *Vendor Tools*: в частности, не выполняется оптимизация трассировки внутренних соединений и временных задержек.

Таким образом, несмотря на то что инновационная станция Altium представляет эффективную интегрированную среду разработки и отладки проектов функциональных узлов на ПЛИС, использование этой среды не отменяет и не подменяет собой разработку проектов в среде САПР ведущих производителей ПЛИС.

Владислав Суходольский  
Бюро ESG,  
эксперт по Altium Designer,  
доцент кафедры  
микроэлектроники  
и технологии радиоаппаратуры  
СПбГЭТУ «ЛЭТИ»



# Форум

«Altium: навстречу российскому пользователю»

Москва,  
8 октября 2013 года

Международный информационно-выставочный центр  
«ИнфоПространство»

Оторвитесь от традиционной рутины и воспользуйтесь новейшими методами проектирования, которые успешно применяются во всем мире.

Работайте с умными комплексными решениями, которые уже завтра станут стандартом.

Расширяйте горизонты ваших проектных возможностей вместе с **Altium Designer!**

Компания Altium Limited является одним из ведущих мировых разработчиков программного обеспечения для проектирования электронных устройств. Такие продукты компании, как Altium Designer (ранее Protel), P-CAD, Tasking и другие, широко известны в мире и зарекомендовали себя как удобные и надежные инструменты разработчиков электроники.

## В ПРОГРАММЕ ФОРУМА\*



Майкл Лайдл  
(Michael Leidel)

Директор Altium по региону EMEA  
«ОБЗОР РЕШЕНИЙ КОМПАНИИ ALTIUM»



Роберт Хаксэл  
(Robert Huxel)

Технический эксперт в области управления данными  
«СИСТЕМА УПРАВЛЕНИЯ ДАННЫМИ ALTIUM VAULT»



Арам Мирказеми  
(Aram Mirkazemi)

Директор по технологиям Altium  
«ИДЕОЛОГИЯ РАЗВИТИЯ ALTIUM DESIGNER 2013»

## ALTIUM: истории успеха российских пользователей

Об опыте работы на платформе расскажут представители крупнейших компаний радиоэлектронной отрасли России.

## «User-zone»: попробуй, оцени, пользуйся!

В течение всего мероприятия будет действовать «User-zone» – площадка с рабочими местами, где вы сможете лично попробовать и оценить все функциональные возможности Altium Designer!

Регистрация на сайте [Altium-forum.ru](http://Altium-forum.ru) и по тел.: +7 (495) 645-86-26, доб. 134

\* В программе мероприятия возможны изменения.

**Altium**