

## Программно-аппаратный модуль для разработки проектов на ПЛИС

Компания Altium дополнила свое семейство макетных плат NanoBoard новой платой, обеспечивающей быструю разработку прототипов электронных устройств на базе Field-Programmable Gate Array (FPGA). Новая плата NanoBoard 3000 является программируемой средой разработки. Помимо аппаратных и программных средств, к плате относятся программные продукты (Intellectual Property, IP), не требующие приобретения отдельной лицензии, и специальная лицензия Soft Design на модуль программного проектирования Altium Designer. Таким образом, пользователь получает все необходимое для быстрой разработки прототипа на FPGA. Отпадает необходимость поиска в Интернете драйверов, периферийных функций и другого программного обеспечения, а также не нужно выполнять кропотливую работу по добавлению этих элементов к функционирующей конструкции.

Олег ИЛЮКИН  
oiilukin@nanocad.ru

Возможность макетирования электронного устройства, отражающего его реальные параметры, на этапе создания принципиальной электрической схемы без затрат времени на проектирование и производство самой печатной платы в период активного развития электроники очень важна.

Благодаря программно-аппаратному комплексу NanoBoard 3000 разработчики электронных устройств, даже не имея опыта работы, могут конструировать в FPGA сложные системы на основе программируемых процессоров. Для этого нет необходимости в подготовке в области программных языков VHDL или Verilog. Разработчики могут использовать уже существующие заготовки печатных плат и свои навыки в системном проектировании для конструирования, испытания и внедрения встраиваемых систем на базе FPGA. IP-библиотеки и интуитивные графические редакторы, являющиеся центральным элементом средства программного проектирования Altium Designer, облегчают добавление в систему процессоров, контроллеров памяти, периферийных блоков и программных стеков. У разработчиков есть все, что нужно для создания на базе FPGA из готовых компонентов встроенных систем нового поколения без написания HDL-кодов. Плата NanoBoard 3000 оснащена (на выбор) FPGA Xilinx Spartan 3AN, Altera Cyclone III, Lattice ECP2. У всех трех вариантов макетных плат матрица FPGA жестко задана.

В макетной плате NanoBoard 3000 реализованы следующие особенности архитектуры:

- Программируемая аппаратная среда позволяет многократно менять конфигурацию системы без каких-либо дополнительных затрат времени и средств.
- Реализована полная синхронизация с Altium Designer — САПР нового поколения компании Altium.
- Поддерживается добавление периферийной платы производства Altium или собственной разработки, что обеспечивает гибкость системы.
- Периферийная плата полностью распознается и конфигурируется.
- Есть возможность обновлять встроенное ПО платы через стандартное USB-соединение, при этом не требуется ни параллельного порта, ни адаптера USB JTAG.

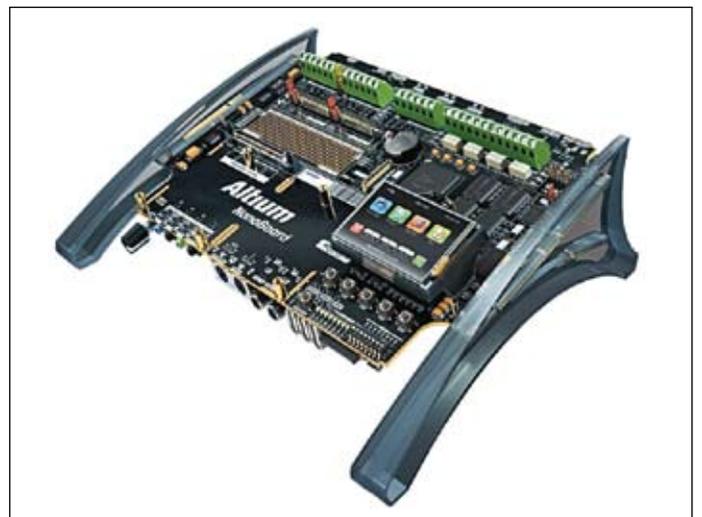
Эта система является синхронизированной, так как при проектировании устройства внутри оболочки Design Explorer разработчик решает сразу несколько задач, связанных в один проект:

- программирование и реализация на ПЛИС;
- разработка проекта на базе ПЛИС;

- проектирование топологии платы;
  - доработка печатной платы до ее производства.
- Универсальность NanoBoard достигается благодаря широчайшему спектру встроенных периферийных устройств, при этом поддерживается работа со многими распространенными интерфейсами: SVGA; RS-232; RS-485; PS/2; 10/100 Fast Ethernet; USB 2.0; S/PDIF; MIDI.

Основная часть NanoBoard — это периферийная плата, которая содержит ту или иную ПЛИС, в которой, собственно, и реализуется созданный проект. К установленной ПЛИС подведены все периферийные устройства и разъемы интерфейсов. Благодаря использованию NanoBoard можно получить прототип устройства, при этом не возникает необходимости создавать физическую модель: проект, реализованный на ПЛИС, позволяет легко и быстро вносить изменения в прототип.

Пользователь описывает проектируемое устройство с помощью принципиальной схемы, в которой наряду с обычными цифровыми компонентами используются особые библиотечные элементы (IP-блоки), соответствующие требуемой периферии, которые подключаются к необходимым узлам принципиальной схемы. При этом для описания устройства можно использовать как стандартные цифровые элементы, которые представлены в обширных поставляемых библи-



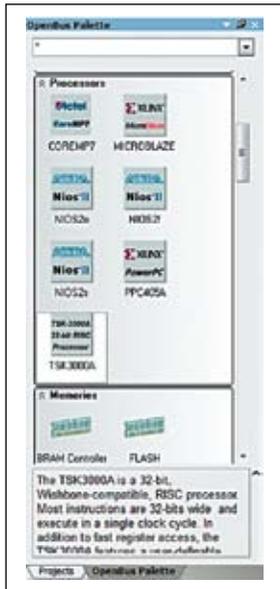


Рис. 1. Панель OpenBus

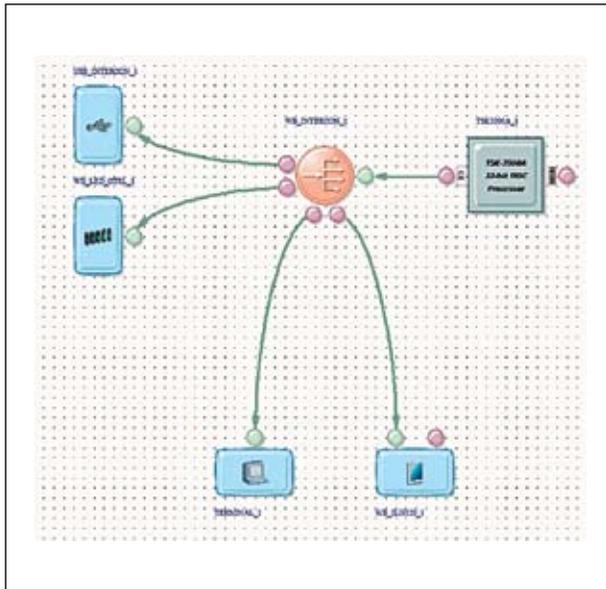


Рис. 2. Промежуточная схема из IP-блоков

отеках, так и встраиваемые VHDL-фрагменты. Аналогично используются и интерфейсы, которые также располагаются на принципиальной схеме, но в отличие от внешней периферии будут размещены внутри ПЛИС при синтезе проекта [1].

Поскольку проект макетируется на базе ПЛИС, возникает необходимость описать распределение ее выводов. Вся информация о распределении выводов ПЛИС описывается в constrain-файлах, которые являются составной частью проекта. При необходимости в одном проекте возможно использование нескольких constrain-файлов, разработчик разделяет между ними, например, описание выводов и требуемые ограничения задержки. Также в одном проекте можно использовать несколько конфигураций для реализации, каждая из которых будет ориентироваться на разные кристаллы и их использование. При необходимости такой файл можно импортировать соответственно из ISE, Quartus или ispLEVER.

При отладке любого устройства на ПЛИС к кристаллу должны подводиться тактирующие сигналы. Для реализации тактирования в NanoBoard предусмотрен синтезатор частот, который подсоединяется к узлам схемы. Его физическое подключение к кристаллу описано в constrain-файле и не требует от разработчика каких-либо действий. Пользователь может задать требуемую частоту синтезатора в диапазоне от 6 до 200 МГц.

Для более детального знакомства с основными этапами работы с NanoBoard создадим проект, в котором задействована USB-клавиатура, с помощью которой будет осуществляться управление светодиодами на передней панели NanoBoard с выводом на TFT-панель.

Создаем проект с самого начала: **File** → **New** → **Project** → **FPGA Project**. После создания проекта его необходимо сохранить:

**Save Project As...** (не забывайте периодически сохранять проект). Далее добавляем в дерево проекта документ **OpenBus** (**Add New to Project** → **OpenBus System Document**).

После создания файла в левой части окна открывается **OpenBus Palette**, в которой есть соединители, микропроцессоры, память и многочисленные периферийные устройства с кратким описанием каждого из элементов (рис. 1). Все устройства в этой панели уже построены на NanoBoard [2]. На данный момент возможность создавать собственные IP-блоки отсутствует, работать можно только с созданными IP-блоками и библиотечными элементами. Следует отметить, что работать с NanoBoard необходимо совместно с Altium Designer.

Далее с помощью готовых IP-блоков начинаем создавать макет будущего устройства. Для начала нужно выбрать микропроцессор. В данном случае выбираем TSK3000A с 32-битной RISC-архитектурой [3]. Для реализации

поставленной ранее задачи необходимы еще другие IP-блоки, такие как USB Interconnect для клавиатуры, LED Controller для подключения светодиодов, Terminal instrument и VGA 32-bit ILI9320 для отображения на TFT-панели, после чего добавляем соединители — Interconnect. Отметим, что красный порт является доминирующим, а зеленый — подчиненным. Получаем часть схемы согласно рис. 2.

Теперь для полноценного функционирования и работы необходимо наличие памяти. Для этого опять нужна панель **OpenBus Palette**. В разделе **Memories** выбираем **SRAM Controller**. Поскольку данную память будут использовать и другие устройства, то в схему надо добавить **Shared Memory NB3000**, при этом настройки памяти оставляем по умолчанию. Далее нужно добавить соединители **Interconnect** и **Arbiter** для определения приоритета обращения к памяти, после чего необходимо соединить все IP-блоки между собой (рис. 3).

Прежде чем перейти к следующей части проекта, нужно сохранить изменения. Следующим шагом будет перенос информации в файл создания принципиальной электрической схемы. Для этого создаем файл **Schematic** (**Add New to Project** → **Schematic**). Открываем этот файл, щелкаем правой кнопкой мыши и нажимаем **Create Sheet Symbol From Sheet or HDL** в подменю **Sheet Actions**. После чего выбираем созданную ранее схему и получаем преобразование файла **OpenBus** в **SchDoc** (рис. 4).

Теперь необходимо произвести то же самое, что и при создании **OpenBus** файла. Единственное отличие лишь в том, что надо использовать специализированную библиотеку FPGA NB3000 Port-Plugin.Intlib, в которой находятся все устройства, входящие в NanoBoard (рис. 5).

Для того чтобы окончательно сформировать файл **SchDoc**, требуется, используя ранее описанную библиотеку, подключить соответствующие УГО к одноименным устройствам. После добавления в схему всех необ-

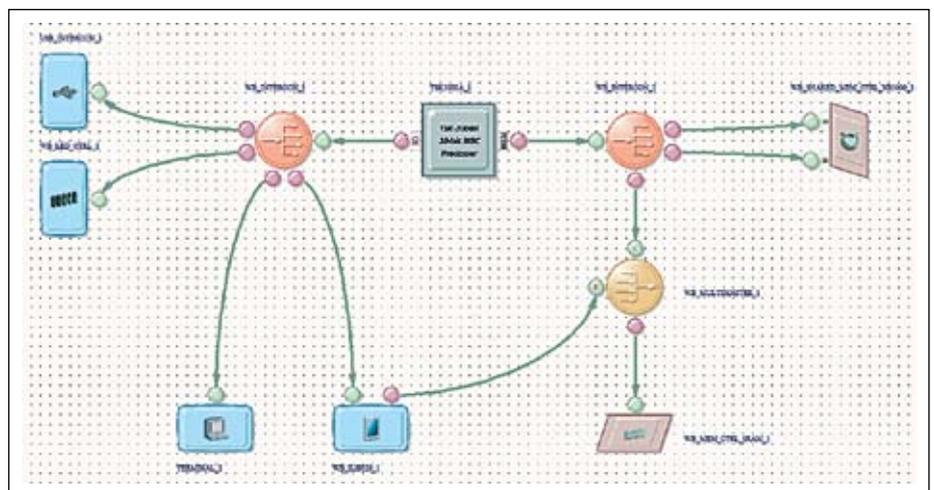


Рис. 3. Готовая схема устройства из IP-блоков

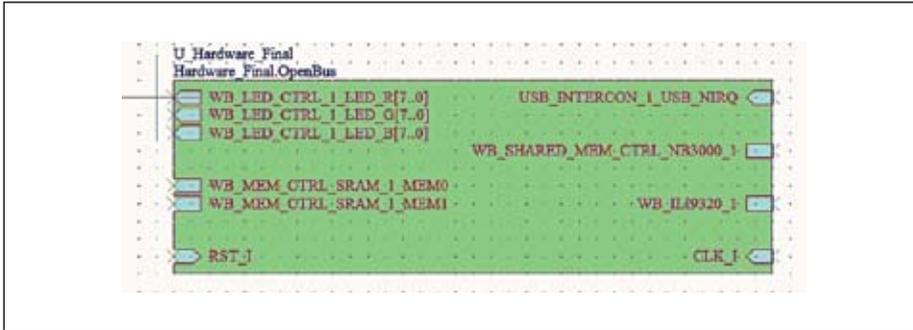


Рис. 4. Создание файла SchDoc

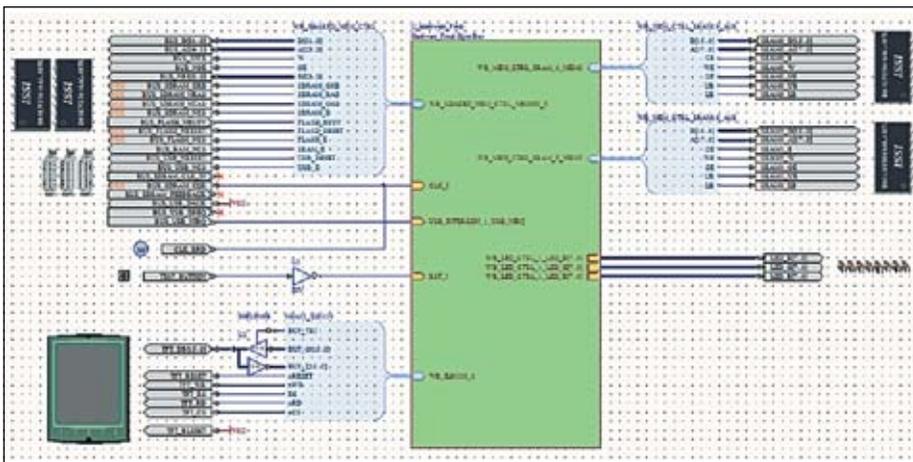


Рис. 6. Готовый к работе файл SchDoc

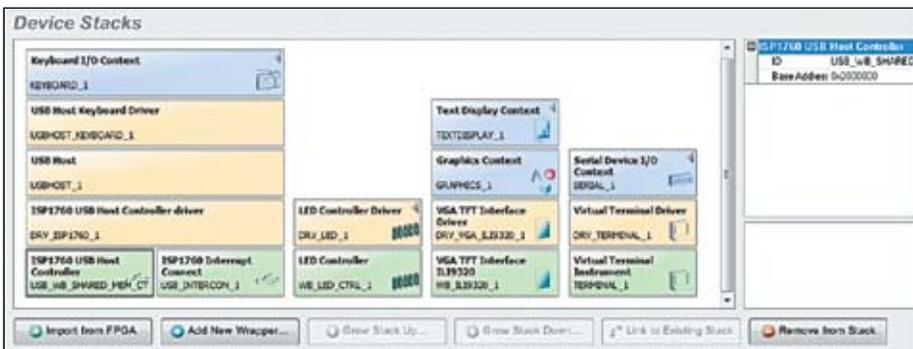


Рис. 7. Окно настройки Device Stack

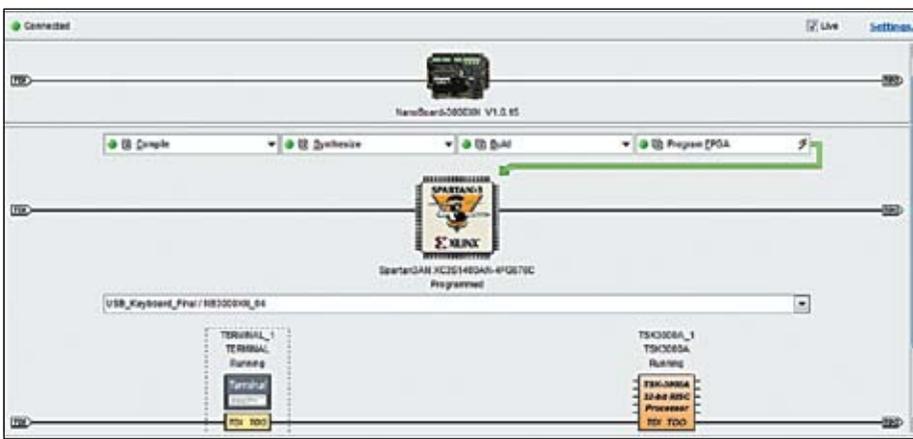


Рис. 8. Синтезирование проекта

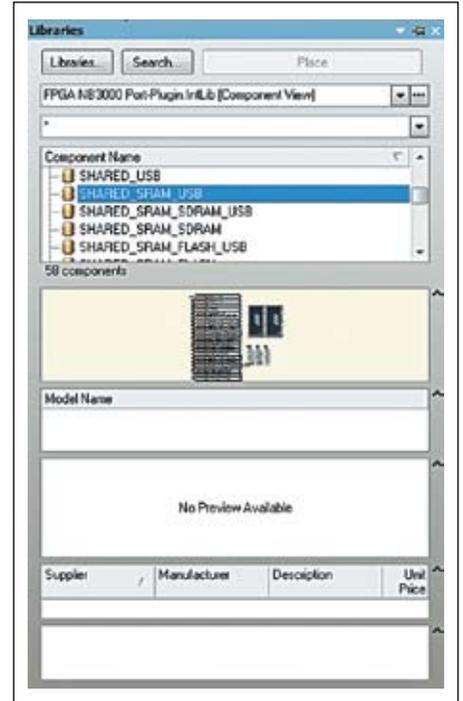


Рис. 5. Библиотека FPGA NB3000 Port-Plugin.Intlib

ходимых элементов получаем завершённую часть проекта аппаратных средств (рис. 6). Теперь нужно провести компиляцию проекта — для выявления ошибок. При верном выполнении предыдущих действий ошибок возникнуть не должно.

Последним этапом перед компиляцией проекта является настройка *Device Stack* (рис. 7). Используя команду *Grow Stack Up*, нужно произвести настройку памяти (USB\_WB\_SHARED\_MEM\_CT), USB-порта (USB\_INTERCON\_1), светодиодов (WB\_LED\_CTRL\_1), TFT-панели (WB\_ILI9320\_1) и виртуального экрана (TERMINAL\_1). Используя команду *Link to Existing Stack*, необходимо объединить USB\_WB\_SHARED\_MEM\_CT и USB\_INTERCON\_1. После того как корректная настройка проведена, можно переходить к последнему шагу — компиляции проекта.

Для компиляции проекта необходимо перейти на вкладку *Devices*. При включении NanoBoard будет отображаться модель и версия программного обеспечения устройства. После этого требуется запустить поочередно каждую вкладку: *Compile*, *Synthesize*, *Build*, *Program FPGA* (рис. 8). Проект готов!

После синтеза проекта во всплывающем окне появится отчет о процентном заполнении кристалла.

Проверить правильность его работы можно с помощью USB-клавиатуры, осуществляя ввод соответствующих команд, отображенных на TFT-панели устройства.

В комплекте поставки приведено множество примеров комплексных устройств, в которых задействована вся периферия. Параллельно можно использовать несколько плат NanoBoard. Для этого используется разъем NanoTalk, кон-



Рис. 9. Модульный корпус для NanoBoard 3000

троллеры которого управляют маршрутизацией сигналов и обеспечивают непрерывную JTAG-связь нескольких плат. Это позволяет одновременно использовать ПЛИС разных производителей, а также проводить тестирование и отладку пользовательской платы. Можно проводить и ручную оптимизацию проектов. Благодаря специальным модульным корпусам (рис. 9) у разработчиков появилась возможность использовать NanoBoard 3000 в качестве готового устройства. ■

### Литература

1. Тархов А. С. Отладка и макетирование цифровых устройств с помощью NanoBoard // EDA Express. 2004. № 10.
2. Altium Designer Help. AR0144 Streamlining Processor-based FPGA Design with the OpenBus System.pdf
3. Altium Designer Help. CR0140 FPGA Processor Resource Usage.pdf