6. Входы/ выходы ядра

В табл. 6-1 приведены все входы/ выходы ядра IP USB20HR.

Табл. 6-1. Входы/ выходы ядра

Тип интерфейса	Имя	Ширина	Направление	Описание
ULPI	Data	8	Bi-Di	Двунаправленная шина данных
	Dir	1	In	Направление шины
	Nxt	1	In	Следующая шина
	Stp	1	Out	Шина остановки
	Phy_clk	1	In	Тактовый сигнал чипа PHY
	Phy_reset_n	1	Out	Сброс чипа PHY
	Phy_cs_n	1	Out	Выбор чипа PHY

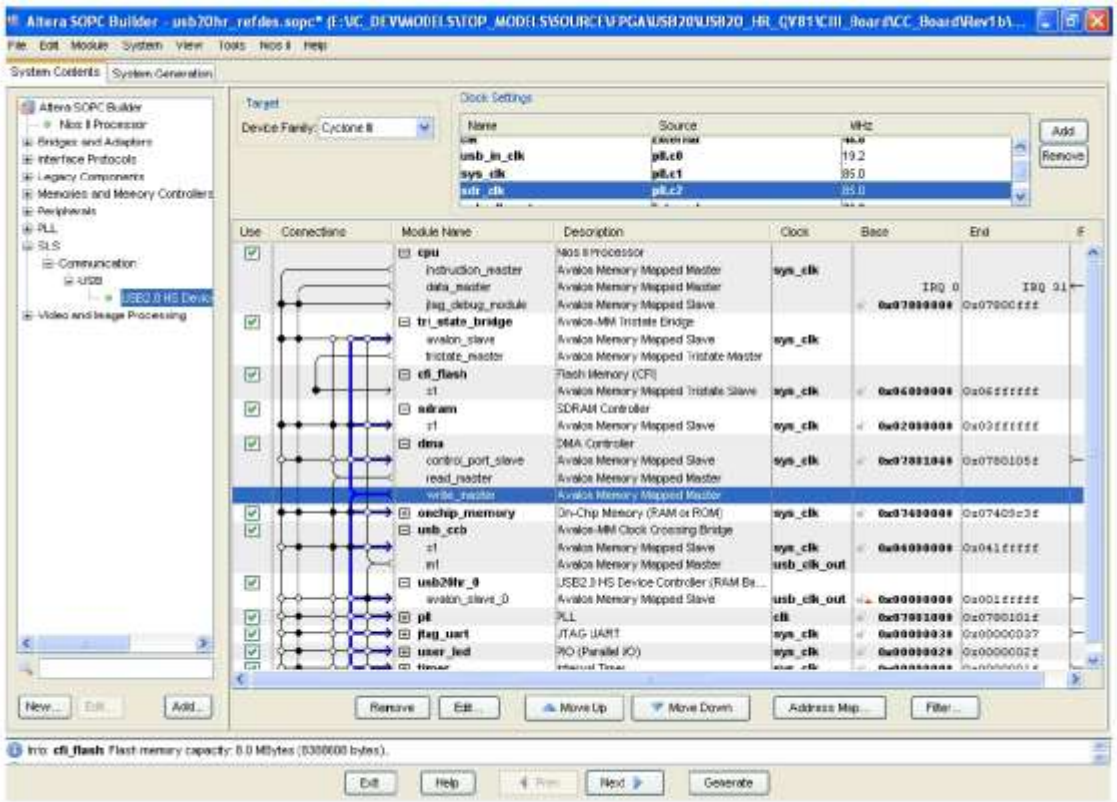
7. Использование IP USB20HR в SOPC Builder

Для использования ядра устройства IP USB20HR в SOPC Builder выполните следующие пункты:

1. Запустите **usb20hr**_<licensetype>_**v**<version #>**.exe**
2. Этим вы автоматически разместите компонент USB20HR внутри оболочки SOPC Builder (См. рис. 7-1).

Где <licensetype> - это тип лицензии ядра, а <version #> - это номер версии IP ядра. licensetype может быть **ocp_eval** или **oc_full**.

Figure 7-1. USB20HR Device IP Core in SOPC Builder



3. Так как ядро - слейв, оно должно иметь Avalon мастер. Этим мастером является процессор Nios. Используя инструмент SOPC Builder, добавьте процессор Nios (должен быть в наборе разработчика). Затем добавьте любую другую периферию, необходимую для конечного проекта, и назначьте базовые адреса и прерывания. За полной информацией обратитесь к главе "Регистры ядра".
4. Запустите поставляемый Си файл, загрузив его в процессор Nios II через IDE или с помощью файла **.elf** в среде Cygwin bash и проследите за поведением ядра с помощью команды *nios-run*.

Ядро может работать от основных прерываний процессора Nios. Это зависит от того, какая операция, чтения или записи, должна выполняться к заданному устройству, что определяется процессором Nios, поскольку он является мастером. Ядро может предоставлять команды для процессора Nios посредством прерываний. Предлагаемый Си файл с примером SLS (посмотрите его в директории //usb20hr/ software/ example/ niosII/ temperale) запускается в заголовке процессора Nios, позволяя ему задавать регистры и декодировать коды операции и команды.