

Общие моменты при разработке на FPGA

При разработке на FPGA разработчику необходимо всегда помнить, что он создает **устройство**, а не пишет **программу**. Т.е., когда создается описание устройства с помощью одного из HDL языков, разрабатывается именно само устройство, его составляющие и связи между ними, а не некая программа на языке программирования. Особенно этот факт нужно держать в голове разработчикам, которые пришли к FPGA от C/C++, Java и т.п.

При разработке цифрового устройства необходимо поддерживать его иерархичность: топмодуль разбивать на составляющие компоненты, эти компоненты на свои составляющие и т.д. В конце концов каждый модуль должен решать одну конкретную задачу. Такой подход упрощает отладку, а также позволяет синтезатору, мапперу и разводчику эффективнее (в смысле количества ресурсов и максимальной тактовой частоты) разводить проект.

Каждый модуль должен иметь только один сигнал тактирования, за исключением модулей, обеспечивающих переход из одного тактового домена в другой. Если какой-либо модуль имеет несколько сигналов тактирования, значит он слишком большой.

Категорически не рекомендуется формирование новых сигналов тактирования из основного с помощью счетчиков (делитель частоты на счетчике). Для создания нового сигнала тактирования из основного необходимо пользоваться специальными встроенными ресурсами FPGA (у Xilinx – это DCM и PLL).

Категорически не рекомендуется пропускать сигнал тактирования через комбинационную логику (gated clock). Сигнал тактирования должен поступать непосредственно на вход тактирования триггера минуя любые комбинационные пути.

Современные FPGA подразумевают реализацию синхронных цифровых устройств. В синхронном устройстве сигналы проходят через комбинационные элементы и триггеры (синхронные). Последние тактируются единым сигналом тактирования. При этом все задержки определяются триггерами, а не комбинационными элементами. Нельзя организовывать петли с обратной связью только из комбинационных элементов.

При реализации комбинационных процессов с условными операторами (if, case) особое внимание уделять завершенности анализируемых условий (проще всего оператор if завершать веткой else, а case – веткой when others). В противном случае будут сгенерированы триггеры-защелки, которые окажут негативное влияние на временные характеристики устройства (синтезатор об этом предупредит, но предупреждение можно, случайно пропустить).

При использовании VHDL желательно избегать применения переменных. Использование переменных может привести к написанию “программы”, а не “разработке устройства”. Это чревато многоуровневыми вложенными конструкциями, которые после синтеза и разводки будут медленными и занимать много ресурсов.

Необходимо избегать сигналов глобального сброса, установки, а также разрешения данных. В противном случае это приведет к переиспользованию соединительных ресурсов FPGA. Т.е. может возникнуть ситуация, когда по количеству ресурсов проект в FPGA поместился, но соединить эти ресурсы не удалось. Речь идет, конечно, о синхронных сигналах.

Не все типы ресурсов конкретной FPGA имеют встроенный сигнал сброса. Если разработчик захочет сбросить такой ресурс, то вход сброса будет эмулирован на дополнительной комбинационной логике, что опять приведет к увеличению занятых ресурсов и снижению максимальной тактовой частоты. Например, часть таблиц истинности FPGA Xilinx могут работать в режиме сдвигающего регистра, но не имеют сигнал сброса. Если разработчик захочет сделать сбрасываемую линию задержки, она займет значительно больше, чем не сбрасываемая.

Не стоит смешивать типы сбросов: синхронный и асинхронный. У современных FPGA вход сброса/установки может работать только в одном из режимов. При желании сделать триггер сбрасываемым и синхронно, и асинхронно, синхронный сброс будет эмулирован на дополнительной таблице истинности.

Активный уровень управления триггерами – единичный. Это значит, что для реализации сброса триггера нулевым уровнем будет задействована дополнительная логика для инверсии сигнала сброса.

Приоритет у управляющих входов триггера разный и строго определенный. Входы в порядке убывания приоритета: сброс/установка, разрешение работы (clock enable), сигнал тактирования. Это значит, что если входу разрешения работы триггера при описании на HDL языке присвоен более высокий приоритет, чем входу сброса, то опять будет задействована дополнительная логика для эмуляции такого функционала.

Перед прошивкой FPGA необходимо в симуляторе выполнить моделирование работы разработанного устройства. Не всегда возможно покрыть тестами весь проект, еще и на разных уровнях иерархии, но основной функционал должен быть проверен до запуска оборудования.

Обязателен статический временной анализ (который автоматически выполняется при сборке прошивки). Он не должен давать ошибок, иначе по результатам функционального моделирования все будет хорошо, а в реальности будут “странные”, не повторяющиеся сбои устройства. Статический временной анализ показывает будет ли работать устройство на заданной тактовой частоте или нет.