

## Способы формирования задержанных копий сигналов.

Необходимость получить задержанные на несколько тактов копии сигналов обычно связана с синхронизацией потоков данных при конвейерной обработке и при формировании сигналов управления конвейером. Следует учитывать, что количество уровней иерархии и степень детализации в виде дробления блоков на более мелкие субблоки определяется разработчиком. На практике, при описании проекта на языках HDL совсем маленькие по функциональности и/или объему кода блоки не выделяются в отдельный иерархический блок (Entity в VHDL или Module в Verilog). Поэтому все написанное в данной заметке применимо и к участкам кода, которые можно было бы выделить в отдельный блок иерархии.

Начнем с простой задачи, изображенной на рис. 1. Данные А и Б формируются на выходе блока №1 одновременно и должны, после обработки в блоках №2,3,4,5 и 6, прийти на вход блока №7 одновременно. Каждый блок имеет на выходе регистр, поэтому задерживает распространение сигнала на 1 такт. Таким образом, обработанные данные Б придут на вход блока №7 на 3 такта позже, чем обработанные данные А. Чтобы сигналы после обработки поступили на вход седьмого блока одновременно необходимо сформировать задержанную на 3 такта копию сигнала А.

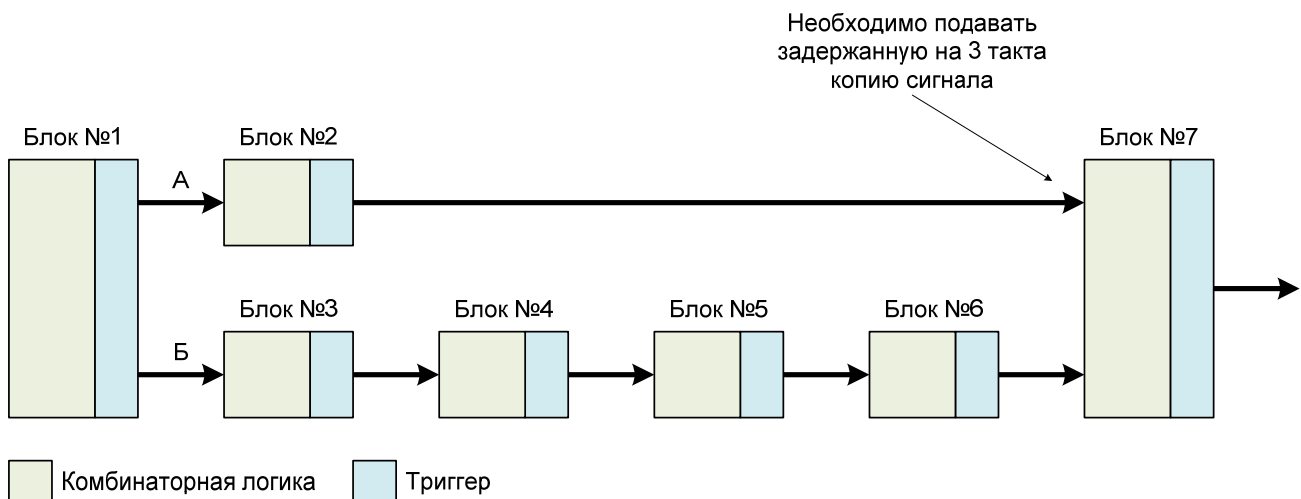


Рис. 1

Задержку можно обеспечить либо поставив триггеры до блока №2, либо поставив их после. Оба места установки задерживающих триггеров функционально эквивалентны. С точки зрения функциональной эквивалентности можно даже поставить часть до, а остальные после. Но с точки зрения экономии ресурсов ПЛИС или ASIC место установки определяется соотношением разрядности входных и выходных сигналов. Если разрядности одинаковы, то все равно где ставить, а если разные, то нужно ставить со стороны меньшей разрядности (см рис. 2).

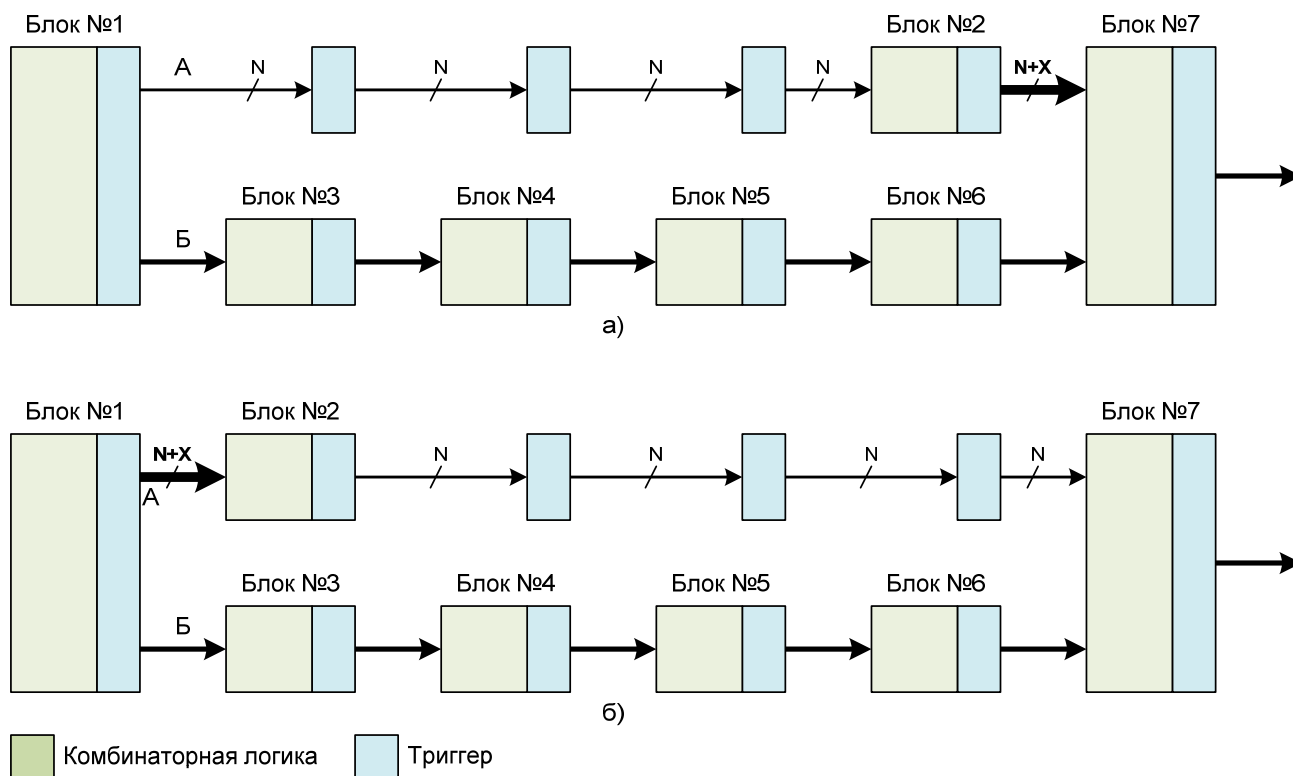


Рис. 2

Теперь рассмотрим задачу, в которой выходные сигналы блока должны идти в качестве входных на несколько блоков с разной задержкой. В этом случае место расположения задерживающих триггеров выбирается не только из соотношения разрядности входных и выходных сигналов. Блок в цепочке задерживающих триггеров не может располагаться дальше того места, которое обеспечивает минимальную из требуемых задержек. Если разрядность входных сигналов больше или равна разрядности выходных, то все триггеры задержки выгоднее располагать за блоком (вариант б рис. 2). В этом случае задержанные копии отводятся с соответствующих задерживающих регистров. Но если разрядность входных сигналов меньше, чем выходных (вариант а рис. 2), то только часть триггеров можно расположить оптимально - т.е. до блока. Такой пример приведен на рис. 3. Сигналы с блока №2 должны придти не только на блок №7 одновременно с сигналами блока №6, но и на блок №8, причем одновременно с сигналами блока №4.

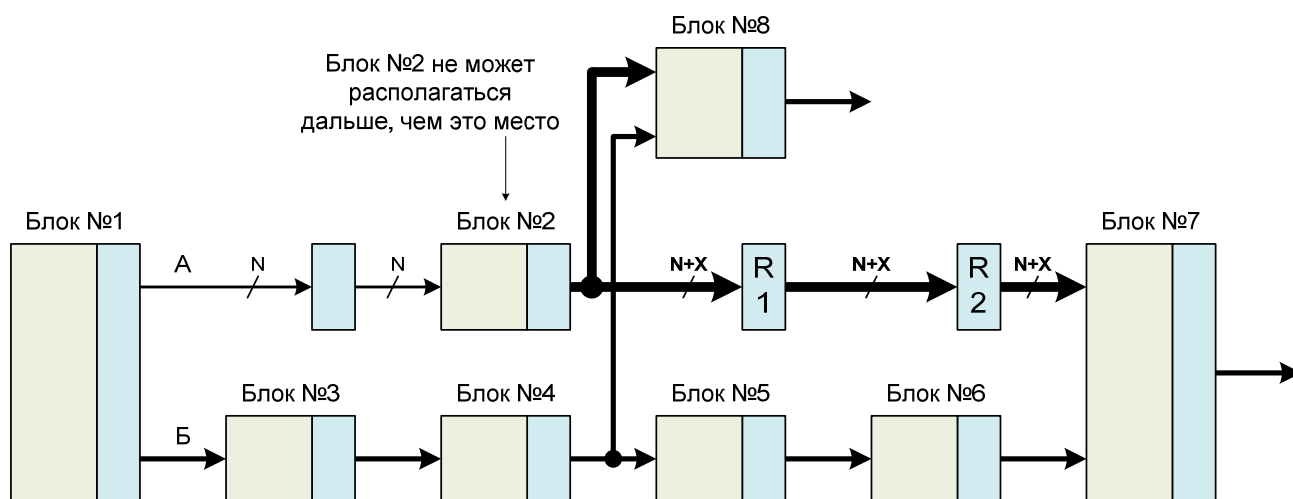


Рис. 3

Другой способ решения - поставить дубликат Блока №2 как показано на рис. 4. Такой способ решения требует более тщательного анализа и будет оправдан только в том случае, если разность площадей, занимаемых регистрами задержки  $r_1$ ,  $r_2$  (рис. 4) и  $R_1$ ,  $R_2$  (рис. 3), будет больше, чем площадь, занимаемая Блоком №2.

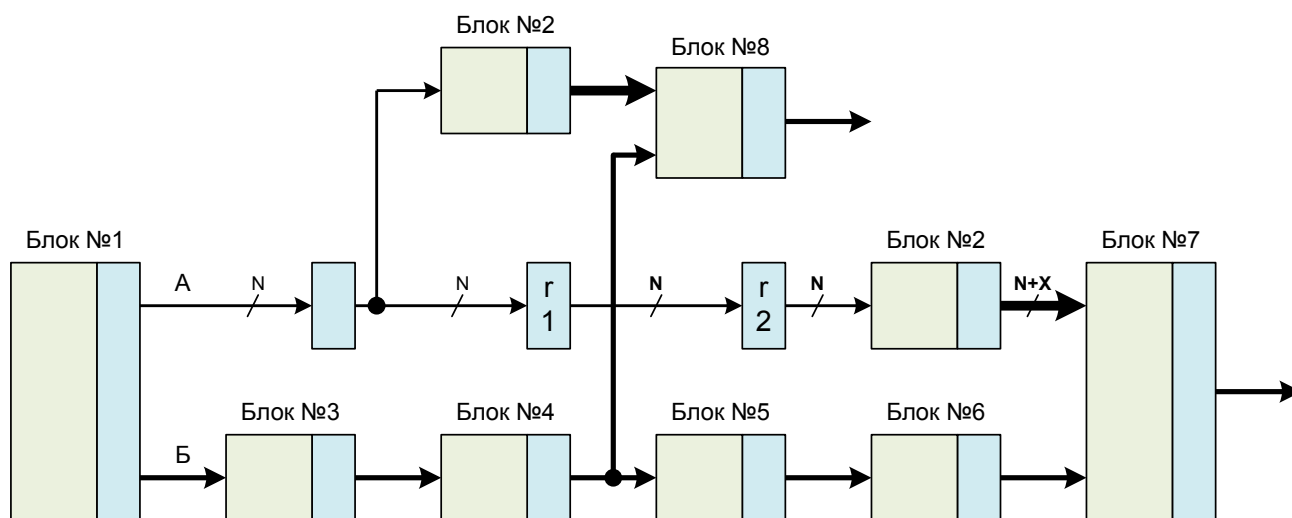


Рис. 4

Обе задачи можно решить еще одним способом, который, правда, очень редко применим. Если исходный сигнал и его копия имеют функциональную связь между собой, то вместо задерживающих триггеров или дубликата исходного блока можно поставить блок, вычисляющий задержанную копию сигнала по значениям исходного. Подробнее этот способ будет рассмотрен в конце.

Важно помнить, что положение блока в цепочке задержки влияет на формирование управляющих сигналов для него и при изменении положения блока необходимо вносить соответствующие корректировки в блок управления. При незначительной разнице разрядности входных и выходных данных при определении оптимального положения блока обработки в цепочке задержки нужно обязательно учитывать изменение площади (причем может иметь место не только увеличение, но и уменьшение площади) блока управления. При вставке дубликата блока изменения площади блока управления оказывают еще большее влияние.

Теперь рассмотрим примеры, когда речь идет о задержанной копии сигнала управления. На рис. 5. На входы конвейера А и Б поступают данные. Каждый третий такт необходимо в обработку брать данные со входа А, а в остальные моменты со входа Б. Далее, к данным со входа А нужно прибавить константу, а из данных со входа Б ее нужно вычесть. Блок управления должен иметь внутри счетчик, по которому на каждый 3-ий такт формируется импульс, переключающий входной мультиплексор. Очевидно, что тот же самый импульс должен подаваться и на арифметический блок, но т.к. мультиплексор задерживает распространение сигнала на 1 такт, то и импульс на арифметический блок должен придти с задержкой. Данный пример полностью соответствует примеру на рис. 3. Поскольку сигнал однобитный, то и единственным оптимальным решением будет добавление задерживающего триггера.

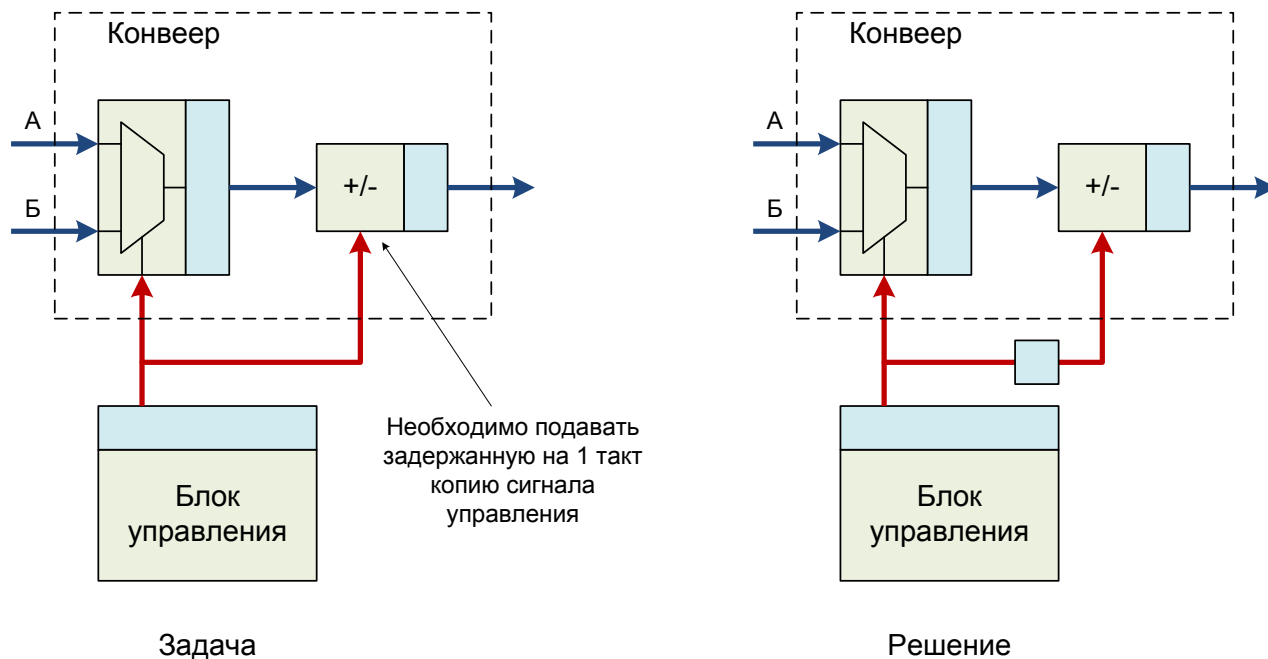


Рис. 5

Теперь рассмотрим другой пример (рис. 6). Возьмем вычислительный модуль, который берет данные из одного блока памяти, три вычислительных блока, выстроенных в конвейер, производят над ними некие действия и результат записывается в другую память по таким же адресам. Что касается задержки Enable, то решение очевидно.

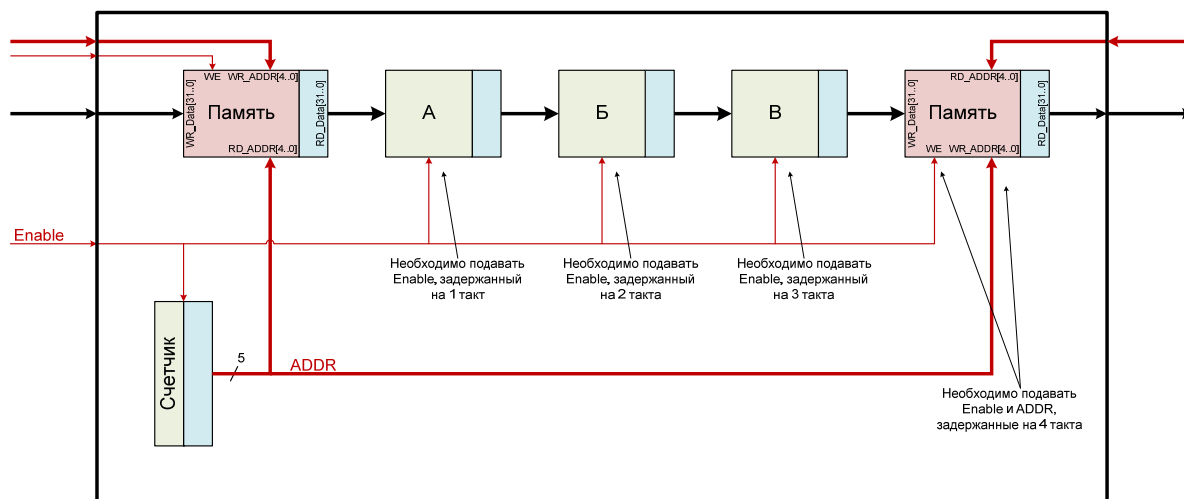
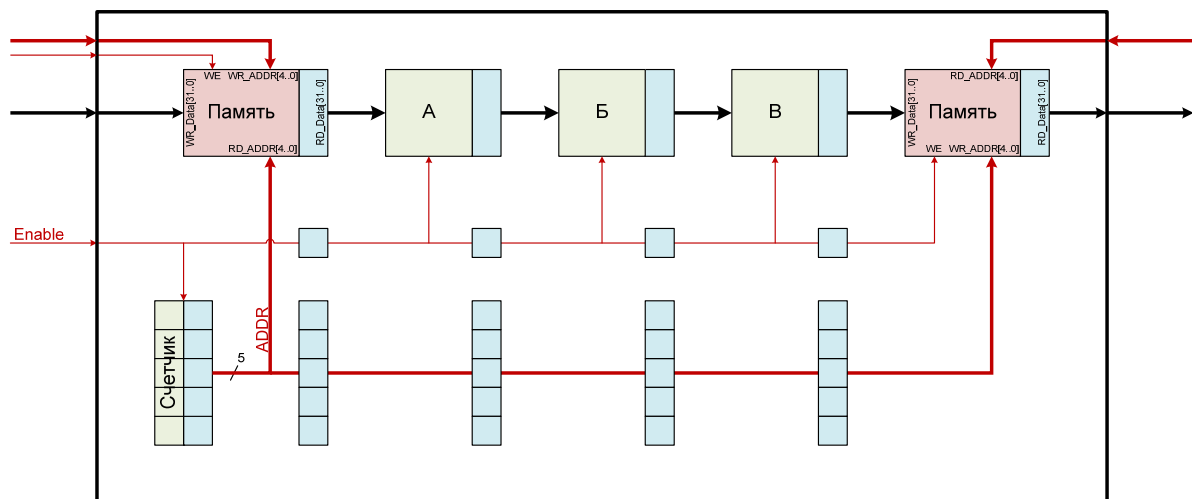
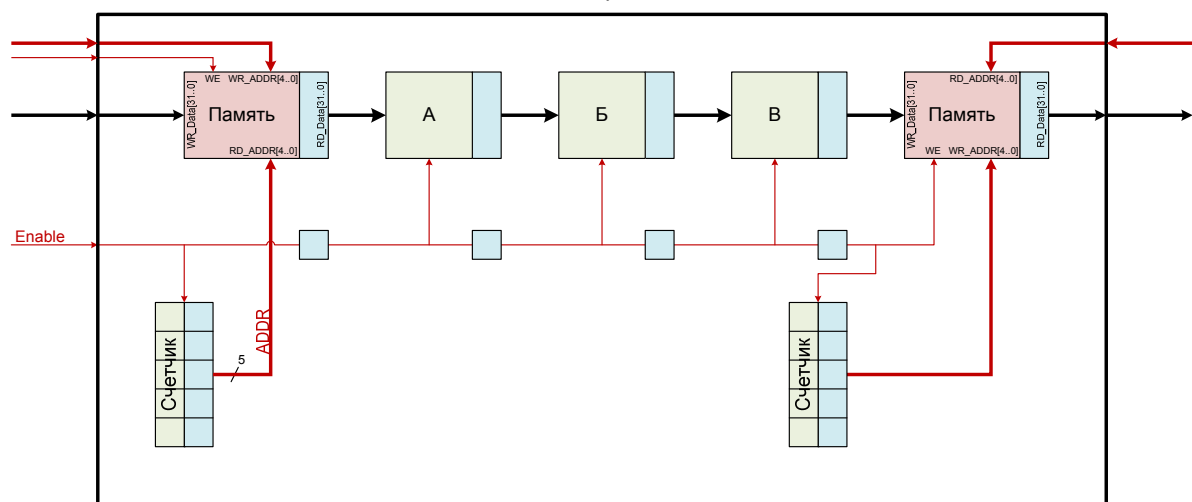


Рис. 6

Поскольку ADDR используется в качестве адреса в обоих блоках памяти, то мы должны сделать выбор между решением, показанным на рис. 3 и решением на рис. 4 (т.е. дублированием). Оба варианта приведены на рис. 7. Управляющая логика нарисована в масштабе.



а)



б)

Рис. 7

Таким образом, в варианте а) управляющая логика занимает 34 квадрата (10 кв. счетчик, 20 кв. регистры задержки ADDR и 4 кв. рег. задер. Enable). В варианте б) управляющая логика занимает 24 квадрата.

На этом примере рассмотрим способ получения задержанной копии, основанный на наличие функциональной связи. В данном случае наблюдается очень простая связь между адресом чтения и записи:  $WR\_ADDR = RD\_ADDR - 3$ . Знак '-' потому, что когда на память - источник подается адрес = 4, на выходе конвеера (на входе памяти - приемника) только появляется результат вычислений над данными, прочитанными по адресу 0. А 3 - это если блок-вычислитель имеет регистры на выходе и результат подается на WR\_ADDR с задержкой. Если регистры не ставить, то  $WR\_ADDR = RD\_ADDR - 4$ . Реализация такого способа показана на рис. 8. В данном случае этот способ занимает такую же площадь, но имеет больше чуть больше "проводов".

Главный недостаток данного способа (он относится ко всем случаям) заключается в том, что как только останавливается генерация исходного сигнала, так сразу же останавливается генерация задержанной копии. Это дополнительно ограничивает область применения.

В примере на рис. 8 снятие сигнала Enable приведет к тому, что адрес записи WR\_ADDR прекратит обновляться за 3 такта до окончания вычислений. Т.е. 3 последних такта все

приходящие данные будут записываться по одному и тому же адресу затирая друг друга. Поэтому схема на рис. 8 может быть использована, когда не важно каким образом происходит завершение работы. Такому случаю соответствует, к примеру, непрерывная обработка данных, где 2-х портовая память используется в режиме FIFO и модуль останавливается только в самом конце работы, когда уже не важно что и куда запишется и запишется ли вообще.

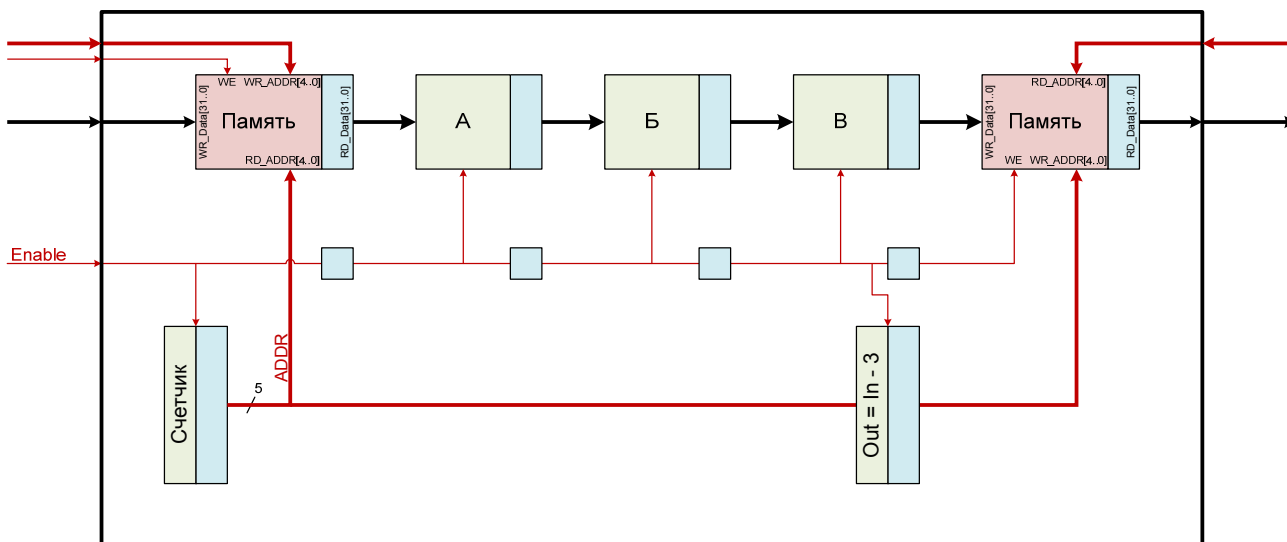


Рис. 8

Еще данный способ может быть использован когда останавливаться конвейер должен весь сразу. Т.е. в нашем примере на рис. 8 снятие сигнала Enable должно прекращать все вычисления и запись результатов в память (или только запись результатов в память). Для этого, правда, схема должна быть доработана как показано на рис 9.

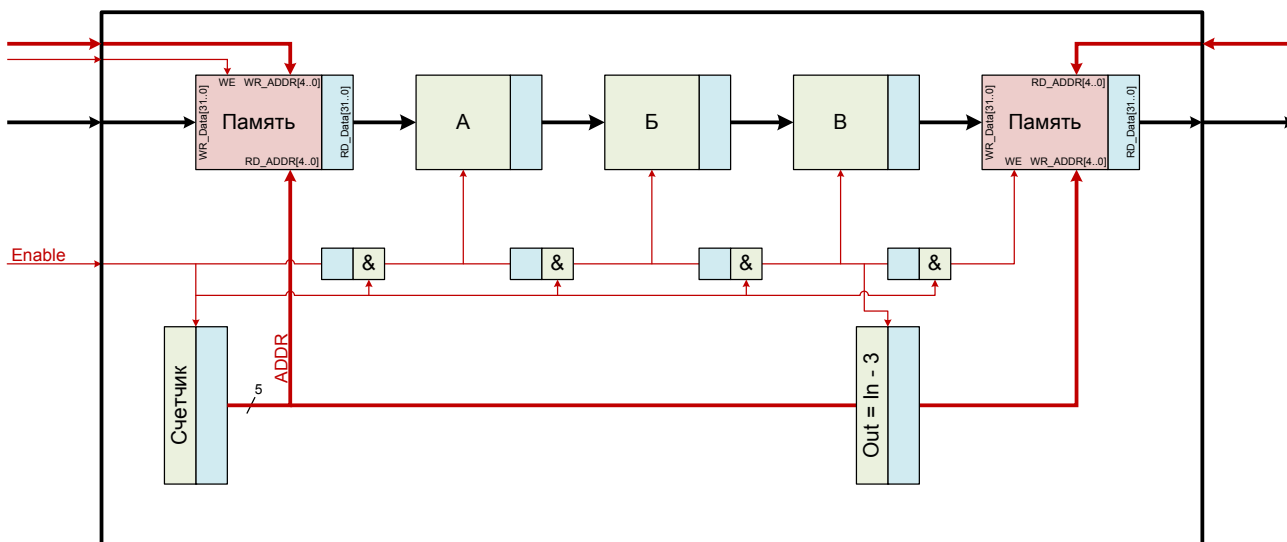


Рис. 9

Замечание по поводу регистров задержки. В большинстве случаев нет необходимости заводить Reset или Enable на такие триггеры. Но тогда, чтобы обеспечить сброс всей системы в исходное и строго определенное состояние нужно удерживать общий сигнал сброса N+1 такт, где N количество тактов задержки в самой длинной цепочке задерживающих триггеров.