

Тема 10. Работа со статической памятью и очередями FIFO

FPGA Spartan 6 содержит много блоков двух портовой статической памяти по 18 кбит и 9 кбит каждая – блочная память. Для разработчиков прикладных программ на ПЛИС это значит, что имеется легко доступный объем ячеек памяти для хранения информации. Легко доступный – это по сравнению с внешней памятью. ПЛИС сама по себе не предназначена для хранения информации, поэтому эта память должна быть использована для сохранения либо некоторых промежуточных значений во время вычислений либо для организации буферов при передаче данных. Например, при обработке потокового видео очередной кадр сначала принимается полностью в ПЛИС, а затем начинаются вычисления и преобразования. Так, этот кадр обычно полностью сохраняется в блочной памяти и хранится там со всеми изменениями в соответствии с алгоритмом. По завершении работы с этим кадром он передается дальше, а его место в блочной памяти занимает следующий кадр.

Короткое описание работы одно портовой статической памяти приведено на Википедии [http://ru.wikipedia.org/wiki/SRAM_\(память\)](http://ru.wikipedia.org/wiki/SRAM_(память)). Одно портовая память – это имеющая один интерфейс доступа к памяти. Существуют также и многопортовая статическая память, имеющая несколько интерфейсов доступа к одним и тем же ячейкам. Широко используется двухпортовая память, когда есть два независимых интерфейса, и можно например читать одновременно любые две ячейки памяти, или в одну писать, а вторую читать, либо писать в разные ячейки. Если производить запись в одну и ту же ячейку памяти одновременно по разным интерфейсам, то будет ошибка – так делать нельзя. Второй порт для доступа обеспечивается дополнительной электроникой (транзисторы на кристалле). Поэтому двухпортовая память дороже чем однопортовая одинаковой емкости. Хорошее и недлинное описание работы однопортовой и двухпортовой статической памяти приведено тут <http://www.cyberguru.ru/hardware/memory/sram-operation.html>.

Рассмотрим 18 кбитный блок двухпортовой статической памяти, реализованный в ПЛИС Рис 1). Оба интерфейса абсолютно идентичны, их описание приведено на рис 2.

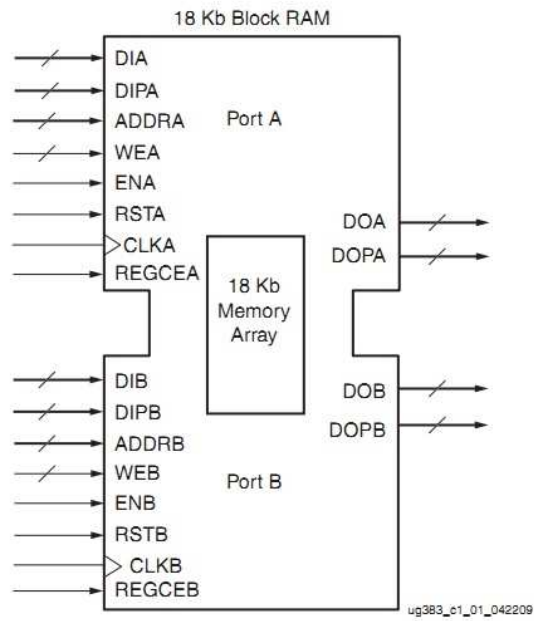


Рис 1.

Port Name	Description
DI[A B] ⁽¹⁾	Data Input Bus
DIP[A B] ⁽¹⁾	Data Input Parity Bus
ADDR[A B]	Address Bus
WE[A B]	Byte-wide Write Enable
EN[A B]	When inactive no data is written to the block RAM and the output bus remains in its previous state.
RST[A B]	Synchronous Set/Reset the output registers (DO_REG = 1).
CLK[A B]	Clock Input
DO[A B] ⁽¹⁾	Data Output Bus
DOP[A B] ⁽¹⁾	Data Output Parity Bus
REGCE[A B]	Output Register Clock Enable

Рис 2.

Блочная память – это синхронный элемент. Работа происходит под управлением тактового сигнала. У каждого из двух портов может быть как одинаковый блок, так и разный. Сигнал EN разрешает какую-либо деятельность с памятью. Если WE = '1' на переднем фронте клона, то происходит запись в память по адресу, который выставлен на линии ADDR (в момент того же фронта клона), значения, присутствующего на входной шине данных DI. Если WE = '0' на переднем фронте клона, то начинается операция чтения по значению адреса ADDR в момент того же фронта клона. Результат будет доступен на выходной шине данных DO спустя небольшую задержку, и, как правило, этот результат считывается с DO в момент следующего фронта клона. Т.е. логика чтения из памяти такая: на первом такте вы выставляете адрес чтения, и на следующем такте получаете результат. Сказанное иллюстрирует рис 3. Входные/выходные шины битов четности, а также сигнала разрешения работы для выходных регистров можно не использовать.

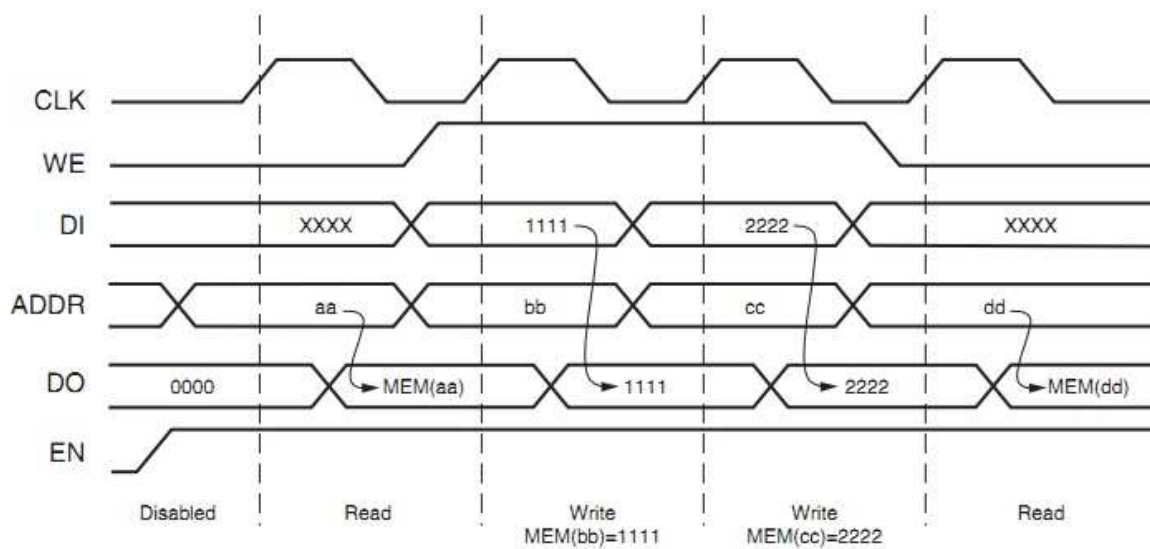


Рис 3. Временная диаграмма работы блочной памяти

Каждый блок (по 9 или 18 кбит) можно сконфигурировать на разную ширину и глубину. Из элементарных блоков по 9 или 18 кбит можно составить массив памяти большей емкости.

Table 4: Block RAM Data Combinations and ADDR Locations

Combinations	Memory Depth	Data Width	Parity Width	Data Input Data Output	ADDR	Total RAM (Kb)
9 Kb Block RAM With and Without Parity						
256 x 32 ⁽¹⁾	256	32	NA	[31:0]	[12:5]	8
256 x 36 ⁽¹⁾	256	32	4	[35:0]	[12:5]	9
512 x 16	512	16	NA	[15:0]	[12:4]	8
512 x 18	512	16	2	[17:0]	[12:4]	9
1K x 8	1024	8	NA	[7:0]	[12:3]	8
1K x 9	1024	8	1	[8:0]	[12:3]	9
2K x 4	2048	4	NA	[3:0]	[12:2]	8
4K x 2	4096	2	NA	[1:0]	[12:1]	8
8K x 1	8192	1	NA	[0:0]	[12:0]	8
18 Kb Block RAM With and Without Parity						
512 x 32	512	32	NA	[31:0]	[13:5]	16
512 x 36	512	32	4	[35:0]	[13:5]	18
1K x 16	1024	16	NA	[15:0]	[13:4]	16
1K x 18	1024	16	2	[17:0]	[13:4]	18
2K x 8	2045	8	NA	[7:0]	[13:3]	16
2K x 9	2048	8	1	[8:0]	[13:3]	18
4K x 4	4096	4	NA	[3:0]	[13:2]	16
8K x 2	8192	2	NA	[1:0]	[13:1]	16
16K x 1	16384	1	NA	[0:0]	[13:0]	16

Рис 4.

Удобный способ использовать блочную память в проекте – это использовать программу Coregen.