

Продолжение. Начало в № 2 '2007

Валерий ЗОТОВ
walerry@km.ru

Разработка базовых компонентов цифровых устройств, реализуемых на базе ПЛИС FPGA фирмы Xilinx®, с помощью генератора параметризованных модулей CORE Generator

Основные характеристики параметризованного модуля FIFO Generator, предназначенного для генерации описаний элементов запоминающих устройств, функционирующих по принципу FIFO

Элементы запоминающих устройств, работающих по принципу «первым вошел — первым вышел» (first-in first-out, FIFO), находят широкое применение в составе различных цифровых систем. В частности, данные элементы памяти используются для организации передачи данных между устройствами, тактируемыми различными сигналами синхронизации, создания высокоскоростных буферных устройств, преобразования разрядности шин данных. Для формирования описаний элементов запоминающих устройств, функционирующих по принципу FIFO, в состав генератора параметризованных модулей CORE Generator включено ядро FIFO Generator. Данный параметризованный модуль относится к группе ядер, предназначенных для подготовки описаний элементов запоминающих устройств *Memories & Storage Elements*. В составе этой группы представлено несколько версий генератора описаний элементов FIFO-памяти. В настоящем разделе подробно рассматривается последняя версия этого ядра (FIFO Generator v4.1), которая обладает наиболее широкими возможностями и поддерживает новые семейства ПЛИС, выпускаемые фирмой Xilinx.

Параметризованный модуль FIFO Generator версии v4.1 предназначен для формирования описаний элементов запоминающих устройств FIFO, которые могут быть реализованы на основе кристаллов следующих семейств: Spartan-2, Spartan-2E, Spartan-3, Spartan-3 XA, Spartan-3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex, QPRO Virtex Rad-Hard, QPRO Virtex

Hi-Rel, Virtex-E, QPRO Virtex-E Military, Virtex-II, Virtex-II PRO, Virtex-4 и Virtex-5. Функциональные возможности рассматриваемой версии генератора элементов FIFO-памяти характеризуются следующими особенностями:

- возможность генерации описаний элементов FIFO-памяти с различной организацией портов записи и чтения данных (соотношение разрядности портов записи и чтения пользователь может выбирать в диапазоне от 1:8 до 8:1);
- широкий диапазон выбора разрядности слова данных в создаваемых описаниях элементов запоминающих устройств, работающих по принципу «первым вошел — первым вышел» (от 1 до 256 разрядов);
- поддержка расширенного диапазона адресного пространства памяти, позволяющего формировать элементы запоминающих

устройств FIFO емкостью от 16 до 4 194 304 слов (фактическая максимальная емкость создаваемого элемента FIFO-памяти определяется объемом физических ресурсов кристалла ПЛИС, выбранного для его реализации);

- возможность формирования описаний запоминающих устройств, функционирующих по принципу FIFO, с общим или отдельным тактированием входного и выходного портов, выполняющих соответственно операции записи и чтения данных;
- поддержка различных способов реализации генерируемых описаний элементов запоминающих устройств, работающих по принципу «первым вошел — первым вышел», (на основе ресурсов блочной памяти ПЛИС Block SelectRAM, распределенной памяти, сдвиговых регистров или встроенных модулей FIFO) по выбору разработчика;

Таблица 6. Функциональные возможности параметризованного модуля FIFO Generator версии v4.1, поддерживаемые элементами запоминающих устройств, работающих по принципу «первым вошел — первым вышел», с отдельным тактированием портов записи и чтения данных

Функциональные возможности параметризованного модуля FIFO Generator v4.1	Вид ресурсов ПЛИС, используемых для реализации генерируемого элемента		
	Блочная память Block SelectRAM	Распределенная память кристалла	Встроенные модули FIFO
Различная разрядность портов записи и чтения данных	Поддерживается	Не поддерживается	Не поддерживается
Использование выходов сигналов (флагов), извещающих о приближении заполнения всех ячеек FIFO-памяти	Поддерживается	Поддерживается	Не поддерживается
Использование выходов сигналов (флагов), информирующих о приближении отсутствия данных, доступных для чтения	Поддерживается	Поддерживается	Не поддерживается
Поддержка совокупности сигналов подтверждения (квитирования)	Поддерживается	Поддерживается	Поддерживается
Наличие выходов счетчика данных	Поддерживается	Поддерживается	Не поддерживается
Наличие дополнительных, программируемых пользователем, выходов сигналов состояния	Поддерживается	Поддерживается	Поддерживается в ограниченных пределах
Применение функции предварительного просмотра данных First-word fall-through (FWFT)	Поддерживается	Поддерживается	Поддерживается только в ПЛИС семейства Virtex-5
Режим синхронного сброса	Не поддерживается	Не поддерживается	Не поддерживается
Режим асинхронного сброса	Поддерживается	Поддерживается	Поддерживается
Определение значений на выходной шине данных для режима сброса	Поддерживается для всех семейств ПЛИС, кроме Spartan-2, Spartan-2E, Virtex-E	Поддерживается	Не поддерживается
Наличие опции коррекции ошибок (Error Correction Checking, ECC)	Поддерживается	Не поддерживается	Поддерживается

Таблица 7. Функциональные возможности параметризованного модуля FIFO Generator версии v4.1, поддерживаемые элементами запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», тактируемых одним сигналом синхронизации

Функциональные возможности параметризованного модуля FIFO Generator v4.1	Вид ресурсов ПЛИС, используемых для реализации генерируемого элемента		
	Блочная память Block SelectRAM	Распределенная память кристалла	Встроенные модули FIFO
Различная разрядность портов записи и чтения данных	Не поддерживается	Не поддерживается	Не поддерживается
Использование выходов сигналов (флагов), извещающих о приближении заполнения всех ячеек FIFO-памяти	Поддерживается	Поддерживается	Не поддерживается
Использование выходов сигналов (флагов), информирующих о приближении отсутствия данных, доступных для чтения	Поддерживается	Поддерживается	Не поддерживается
Поддержка совокупности сигналов подтверждения (квитирования)	Поддерживается	Поддерживается	Поддерживается
Наличие выходов счетчика данных	Поддерживается	Поддерживается	Не поддерживается
Наличие дополнительных, программируемых пользователем, выходов сигналов состояния	Поддерживается	Поддерживается	Поддерживается в ограниченных пределах
Применение функции предварительного просмотра данных First-word fall-through (FWFT)	Не поддерживается	Не поддерживается	Поддерживается только в ПЛИС семейства Virtex-5
Режим синхронного сброса	Поддерживается	Поддерживается	Не поддерживается
Режим асинхронного сброса	Поддерживается	Поддерживается	Поддерживается
Определение значений на выходной шине данных для режима сброса	Поддерживается для всех семейств ПЛИС, кроме Spartan-2, Spartan-2E, Virtex-E	Поддерживается	Не поддерживается
Наличие опции коррекции ошибок (Error Correction Checking, ECC)	Поддерживается	Не поддерживается	Поддерживается

- выборочное использование в создаваемых элементах FIFO-памяти режима синхронного или асинхронного сброса;
- наличие опции коррекции ошибок (Error Correction Checking, ECC) в элементах, реализуемых на базе блочной памяти ПЛИС Block SelectRAM или встроенных модулей FIFO в кристаллах семейства Virtex-5;
- возможность формирования описаний элементов запоминающих устройств FIFO с функцией First-word fall-through (FWFT), позволяющей осуществлять предварительный просмотр следующего слова данных, хранящегося в этом элементе памяти, без выполнения операции чтения;
- применение (по выбору пользователя) в создаваемых элементах FIFO-памяти программируемых выходов сигналов состояния (флагов), информирующих о достижении заданных границ установленного диапазона количества слов данных в этих запоминающих устройствах;
- возможность использования в формируемых запоминающих устройствах FIFO выходов счетчика слов данных, записанных в память и доступных для чтения;
- поддержка организации функционирования генерируемых элементов FIFO-памяти с использованием совокупности сигналов подтверждения (квитирования);
- возможность применения в составе создаваемых запоминающих устройств FIFO встроенных выходных регистров по выбору разработчика;
- наличие в формируемых элементах FIFO-памяти выходов сигналов состояния (флагов), сигнализирующих о записи максимально возможного количества слов данных (заполнении всех ячеек этих запоминающих устройств) или считывании (извлечении) всей записанной ранее информации;
- поддержка выборочного использования в создаваемых запоминающих устройствах FIFO

дополнительных выходов сигналов состояния (флагов), извещающих о приближении заполнения всех ячеек памяти или считывания всей записанной ранее информации. Следует обратить внимание на то, что состав поддерживаемых функциональных возможностей зависит от типа создаваемого элемента FIFO-памяти и вида ресурсов ПЛИС, выбранных для его реализации. В частности, описания элементов запоминающих устройств

с различной организацией портов записи и чтения данных, формируемые с помощью параметризованного модуля FIFO Generator версии v4.1, могут быть реализованы только на базе ресурсов блочной памяти ПЛИС Block SelectRAM. При этом для синхронизации портов записи и чтения данных должны использоваться различные тактовые сигналы. Более подробные сведения о функциональных возможностях рассматриваемого ядра, поддерживаемых запоминающими устройствами FIFO с отдельным тактированием портов записи и чтения данных, приведены в таблице 6.

Для элементов FIFO-памяти, синхронизируемых одним тактовым сигналом, информация о поддерживаемых функциональных возможностях параметризованного модуля FIFO Generator версии v4.1 отражена в таблице 7.

Элементы запоминающих устройств FIFO, описания которых формируются с помощью рассматриваемого параметризованного модуля, отличаются оптимизированной архитектурой, позволяющей добиться высокой производительности в сочетании с минимальным объемом используемых ресурсов кристалла. При этом максимально возможная производительность генерируемого элемента FIFO-памяти зависит не только от выбранного семейства ПЛИС и конкретного типа кристалла, но и от использования дополнительных возможностей, предоставляемых пара-

Таблица 8. Основные характеристики различных типов элементов запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», создаваемых с помощью параметризованного модуля FIFO Generator версии v4.1 для реализации на базе кристаллов семейств Virtex-II, Virtex-4, Virtex-5

Тип элемента FIFO-памяти	Информационная емкость	Семейство ПЛИС	Максимальная тактовая частота, МГц	Объем используемых ресурсов ПЛИС				
				Таблицы преобразования LUT	Триггеры	Блочная память Block SelectRAM	Сдвиговые регистры	Распределенная память
Элементы FIFO-памяти с одним общим тактовым сигналом и реализацией на основе блочной памяти	512 слов × 16 разрядов	Virtex-5	350	62	45	1	0	0
		Virtex-4	200	64	43	1	0	0
		Virtex-II	175	64	43	1	0	0
	4096 слов × 16 разрядов	Virtex-5	350	84	57	2	0	0
		Virtex-4	225	80	55	4	0	0
		Virtex-II	175	80	55	4	0	0
Элементы FIFO-памяти с одним общим тактовым сигналом и реализацией на основе распределенной памяти	64 слов × 16 разрядов	Virtex-5	475	48	61	0	0	32
		Virtex-4	250	49	83	0	0	128
		Virtex-II	225	49	83	0	0	128
	512 слов × 16 разрядов	Virtex-5	375	85	80	0	0	256
		Virtex-4	200	135	438	0	0	1024
		Virtex-II	150	135	438	0	0	1024
Элементы FIFO-памяти с двумя независимыми тактовыми сигналами и реализацией на основе блочной памяти	512 слов × 16 разрядов	Virtex-5	350	105	124	1	0	0
		Virtex-4	250	103	105	1	0	0
		Virtex-II	200	103	105	1	0	0
	4096 слов × 16 разрядов	Virtex-5	350	147	163	2	0	0
		Virtex-4	250	134	138	4	0	0
		Virtex-II	175	134	138	4	0	0
Элементы FIFO-памяти с двумя независимыми тактовыми сигналами и реализацией на основе распределенной памяти	64 слов × 16 разрядов	Virtex-5	475	76	104	0	0	32
		Virtex-4	325	70	123	0	0	128
		Virtex-II	225	70	123	0	0	128
	512 слов × 16 разрядов	Virtex-5	350	126	148	0	0	256
		Virtex-4	225	173	239	0	0	1024
		Virtex-II	150	173	239	0	0	1024
Элементы FIFO-памяти, реализуемые на основе сдвиговых регистров	64 слов × 16 разрядов	Virtex-5	450	59	50	0	32	32
		Virtex-4	200	68	43	0	64	64
		Virtex-II	150	68	43	0	64	64
	512 слов × 16 разрядов	Virtex-5	275	159	67	0	256	256
		Virtex-4	200	312	82	0	512	512
		Virtex-II	150	312	82	0	512	0

Таблица 9. Основные характеристики различных типов элементов запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», создаваемых с помощью параметризованного модуля FIFO Generator версии v4.1 для реализации на базе встроенных модулей FIFO кристаллов семейства Virtex-5

Тип элемента FIFO-памяти	Информационная емкость	Режим чтения данных	Максимальная тактовая частота, МГц	Объем используемых ресурсов ПЛИС		
				Таблицы преобразования LUT	Триггеры	Встроенные модули FIFO36
Элементы FIFO-памяти с одним общим тактовым сигналом	512 слов ×72 разрядов	Стандартный	500	0	2	1
		С функцией FWFT	400	2	4	1
	16 К слов ×8 разрядов	Стандартный	350	10	6	4
		С функцией FWFT	375	13	10	4
Элементы FIFO-памяти с одним общим тактовым сигналом и совокупностью сигналов подтверждения	512 слов ×72 разрядов	Стандартный	500	2	6	1
		С функцией FWFT	400	4	6	1
	16 К слов ×8 разрядов	Стандартный	350	12	12	4
		С функцией FWFT	375	16	13	4
Элементы FIFO-памяти с двумя независимыми тактовыми сигналами	512 слов ×72 разрядов	Стандартный	500	0	2	1
		С функцией FWFT	500	0	2	1
	16 К слов ×8 разрядов	Стандартный	375	6	2	4
		С функцией FWFT	375	6	2	4
Элементы FIFO-памяти с двумя независимыми тактовыми сигналами и совокупностью сигналов подтверждения	512 слов ×72 разрядов	Стандартный	500	2	6	1
		С функцией FWFT	500	2	4	1
	16 К слов ×8 разрядов	Стандартный	350	8	8	4
		С функцией FWFT	350	9	5	4

Таблица 10. Основные характеристики различных типов элементов запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», создаваемых с помощью параметризованного модуля FIFO Generator версии v4.1 для реализации на базе встроенных модулей FIFO кристаллов семейства Virtex-4

Тип элемента FIFO-памяти	Информационная емкость	Соотношение частот тактовых сигналов блоков чтения и записи данных	Максимальная тактовая частота, МГц	Объем используемых ресурсов ПЛИС			
				Таблицы преобразования LUT	Триггеры	Распределенная память	Встроенные модули FIFO
Встроенные элементы запоминающих устройств FIFO без выходов сигналов подтверждения	512 слов ×36 разрядов	WR_CLK ≥ RD_CLK	375	110	129	72	1
		RD_CLK > WR_CLK	400	92	115	72	1
Встроенные элементы запоминающих устройств FIFO с поддержкой совокупности сигналов подтверждения	512 слов ×36 разрядов	WR_CLK ≥ RD_CLK	375	113	134	72	1
		RD_CLK > WR_CLK	400	95	120	72	1

метризованным модулем FIFO Generator версии v4.1. Для создания высокоскоростных запоминающих устройств, работающих по принципу «первым вошел — первым вышел», целесообразно использовать кристаллы семейств Virtex-II, Virtex-4 и Virtex-5. В таблице 8 представлена сравнительная информация о быстродействии и объеме необходимых ресурсов ПЛИС для различных типов элементов FIFO-памяти, реализуемых на базе кристаллов этих семейств. В данных запоминающих устройствах не используются расширенные возможности, поддерживаемые параметризованным модулем FIFO Generator версии v4.1.

Встроенные модули FIFO, конфигурируемые на базе блочной памяти Block SelectRAM и специальной управляющей логики в кристаллах семейства Virtex-5, предоставляют возможность разработки элементов FIFO-памяти достаточно большой емкости с максимальной тактовой частотой, достигающей 500 МГц. Таблица 9 содержит сведения о быстродействии и объеме необходимых ресурсов ПЛИС для различных типов элементов запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», создаваемых с помощью параметризованного модуля FIFO Generator версии v4.1 для реализации на основе указанных встроенных модулей.

В параметризованном модуле FIFO Generator версии v4.1 предусмотрена также возможность эффективного использования встроенных модулей FIFO кристаллов семейства Virtex-4, конфигурируемых на основе блочной памяти Block SelectRAM и специальной управляющей логики, в создаваемых описаниях запоминающих устройств, функцио-

нирующих по принципу «первым вошел — первым вышел». Информация, приведенная в таблице 10, дает представление о быстродействии и объеме необходимых различных ресурсов ПЛИС для элементов FIFO-памяти, реализуемых на базе соответствующих встроенных модулей кристаллов этого семейства.

Формирование описаний элементов запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», с одним сигналом синхронизации портов записи и чтения данных на основе параметризованного модуля FIFO Generator версии v4.1 с помощью средств CORE Generator

Быстро подготовить описание элемента запоминающего устройства, работающего по принципу «первым вошел — первым вышел», с требуемой архитектурой позволяет «мастер» настройки параметров ядра FIFO Generator версии v4.1, который включает в себя шесть диалоговых панелей. Стартовая диалоговая панель этого «мастера», вид которой показан на рис. 138, предназначена для определения названия создаваемого элемента, выбора типа генерируемого запоминающего устройства FIFO и определения ресурсов кристалла для его реализации. В этой же диалоговой панели отображается информация о дополнительных возможностях параметризованного модуля FIFO Generator версии v4.1, которые доступны для каждого варианта архитектуры элементов памяти, работающих по принципу «первым вошел — первым вышел».

Для выбора типа архитектуры формируемого элемента FIFO-памяти и вида ресурсов ПЛИС, используемых для его реализации,

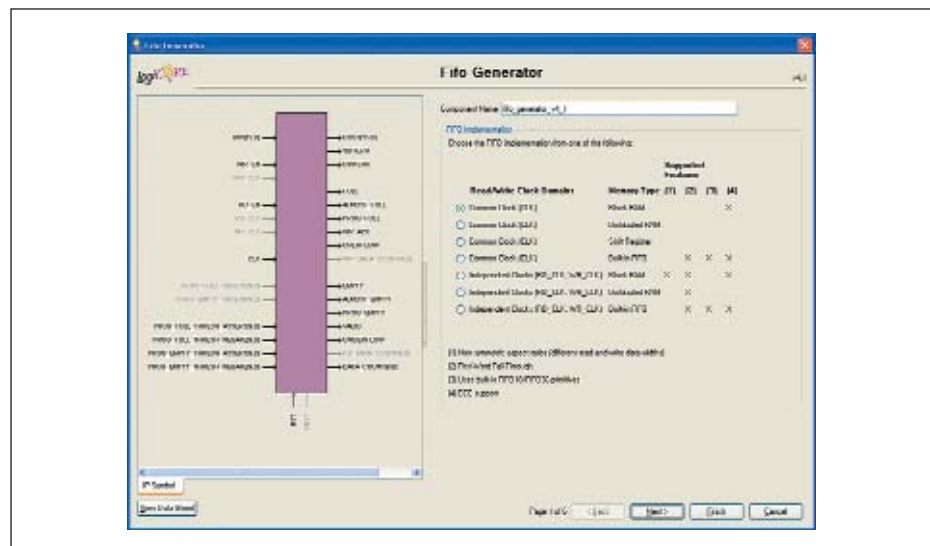


Рис. 138. Вид стартовой диалоговой панели «мастера» настройки параметров генератора элементов памяти, работающих по принципу «первым вошел — первым вышел», FIFO Generator версии v4.1

следует воспользоваться группой кнопок с зависимой фиксацией, которые расположены во встроенной панели *FIFO Implementation*. Чтобы сгенерировать описание элемента запоминающего устройства, работающего по принципу «первым вошел — первым вышел», реализуемого на основе модулей блочной памяти кристалла Block SelectRAM, в котором используется один сигнал синхронизации, следует нажать кнопку *Common Clock (CLK)* — *Block RAM*. Структурное представление архитектуры элементов FIFO-памяти данного типа показано на рис. 139.

Если необходимо подготовить описание элемента запоминающего устройства FIFO, предназначенного для реализации на базе ресурсов распределенной памяти ПЛИС, в котором порты записи и чтения данных тактируются одним сигналом синхронизации, то нужно переключить в нажатое положение кнопку *Common Clock (CLK)* — *Distributed RAM*. В этом случае структура формируемых элементов соответствует архитектуре, изображенной на рис. 140.

Для генерации описания элемента FIFO-памяти с одним тактовым сигналом, которое должно быть реализовано на основе сдвиговых регистров, необходимо зафиксировать в нажатом состоянии кнопку *Common Clock (CLK)* — *Shift Register*. Создаваемые при этом элементы имеют архитектуру, вид которой показан на рис. 141.

При нажатой кнопке *Common Clock (CLK)* — *Built-in FIFO* создается описание элемента запоминающего устройства, функционирующего по принципу «первым вошел — первым вышел», в котором применяется один сигнал синхронизации портов чтения и записи данных, для последующей реализации на базе встроенных модулей FIFO в ПЛИС семейств Virtex-4 и Virtex-5. На рис. 142 приведено структурное представление архитектуры элементов данного типа FIFO-памяти.

Вторая диалоговая панель «мастера» настройки параметров ядра FIFO Generator версии v4.1 используется для выбора режима чтения данных, определения информационной емкости формируемого элемента FIFO-памяти, необходимости применения функции коррекции ошибок и включения выходных регистров в состав создаваемого запоминающего устройства. Вид этой диалоговой панели изображен на рис. 143.

Режим чтения данных в формируемом элементе FIFO-памяти указывается с помощью группы кнопок с зависимой фиксацией, которая расположена во встроенной панели *Read Mode*. При подготовке описаний элементов запоминающих устройств FIFO, в которых используется один сигнал синхронизации портов записи и чтения данных, в большинстве случаев доступен только стандартный режим чтения (в нажатом состоянии постоянно зафиксирована кнопка *Standard FIFO*). Исключением является вариант запоминающего устройства, реализуе-

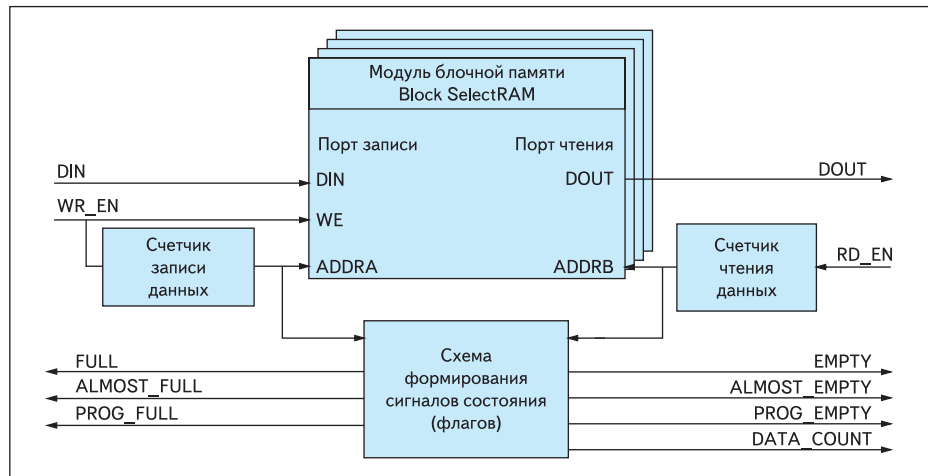


Рис. 139. Структурное представление архитектуры элементов запоминающих устройств FIFO, реализуемых на основе модулей блочной памяти кристалла Block SelectRAM, в которых используется один сигнал синхронизации портов записи и чтения данных

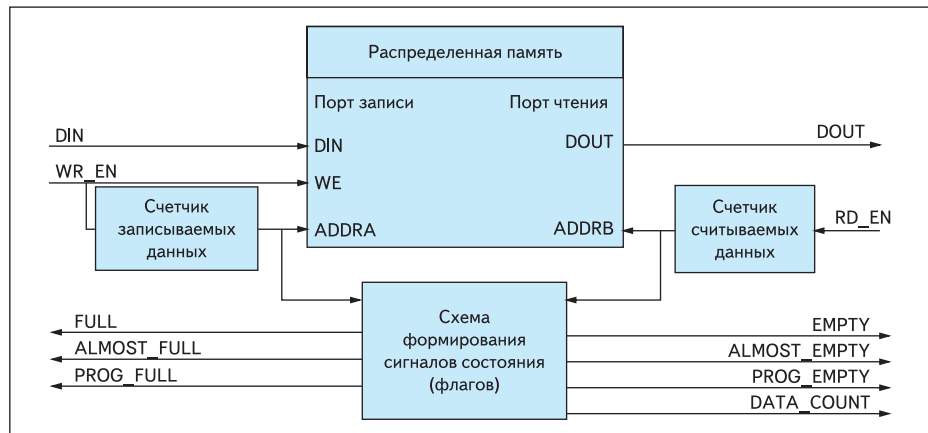


Рис. 140. Структурное представление архитектуры элементов запоминающих устройств FIFO, реализуемых на базе ресурсов распределенной памяти кристалла, в которых используется один сигнал синхронизации для портов записи и чтения данных

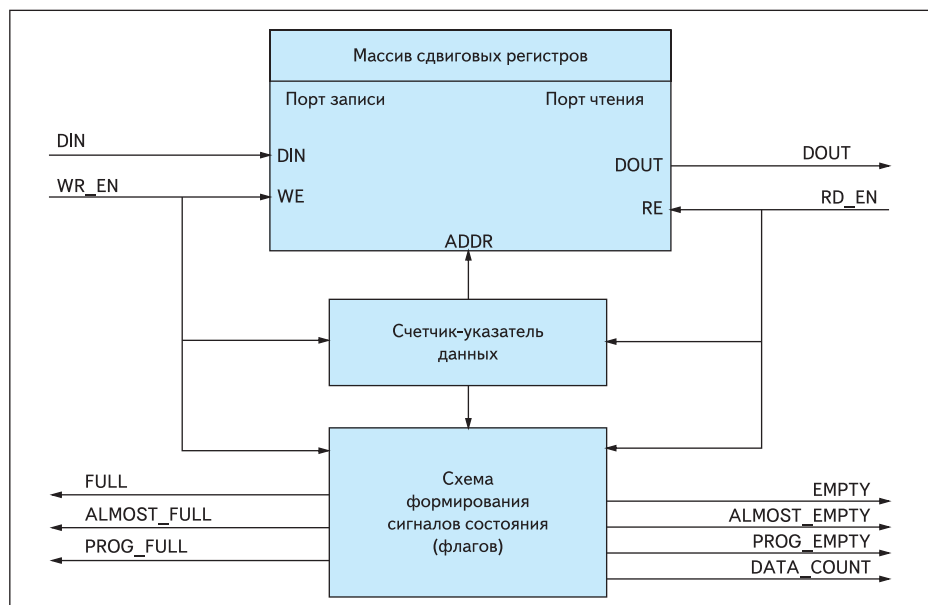


Рис. 141. Структурное представление архитектуры элементов запоминающих устройств FIFO, реализуемых на основе сдвиговых регистров кристалла, в которых используется один сигнал синхронизации для портов записи и чтения данных

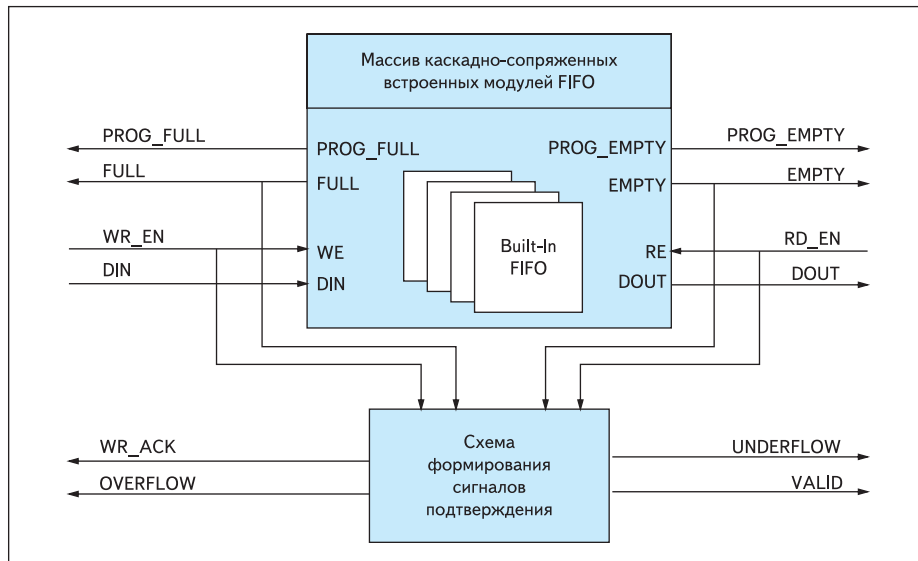


Рис. 142. Структурное представление архитектуры элементов запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», реализуемых на базе встроенных модулей FIFO, в которых используется один сигнал синхронизации для портов записи и чтения данных

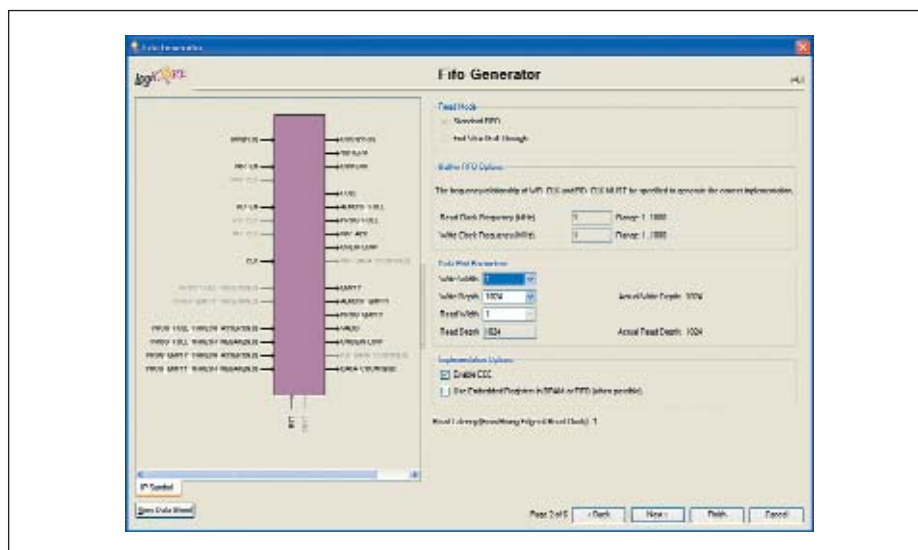


Рис. 143. Вид второй диалоговой панели «мастера» настройки параметров генератора элементов памяти, работающих по принципу «первым вошел — первым вышел», Fifo Generator версии v4.1

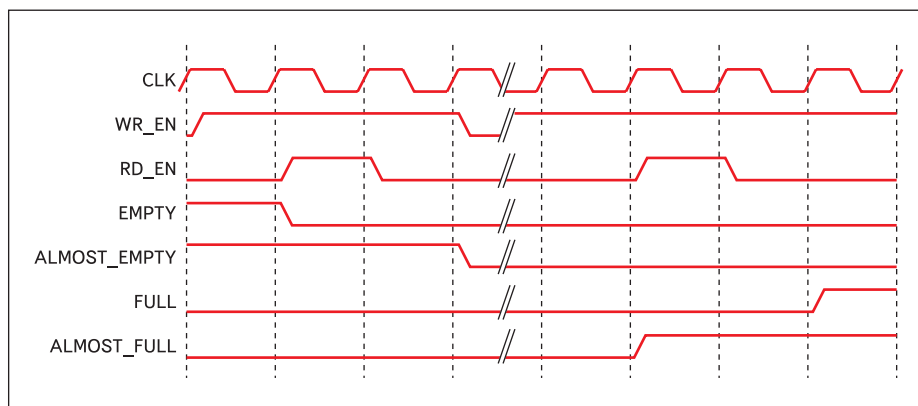


Рис. 144. Временные диаграммы, демонстрирующие последовательность выполнения операций записи и чтения данных в запоминающих устройствах FIFO, в которых используется один сигнал синхронизации для входного и выходного портов

мый на базе встроенных модулей FIFO в ПЛИС семейства Virtex-5. В таких элементах может применяться режим чтения информации с функцией предварительного просмотра данных FWFT. Для поддержки этого режима в формируемых элементах FIFO-памяти необходимо нажать кнопку *First-word Fall-Through FIFO*. На рис. 144 приведены временные диаграммы, демонстрирующие последовательность выполнения операций записи и чтения данных в запоминающих устройствах FIFO, в которых используется один сигнал синхронизации для входного и выходного портов. Представленные временные диаграммы соответствуют стандартному режиму чтения данных.

Для определения информационной емкости и разрядности портов записи и чтения данных генерируемого запоминающего устройства, работающего по принципу «первым вошел — первым вышел», следует воспользоваться полями редактирования, которые находятся во встроенной панели *Data Port Parameters* (рис. 143). Разрядность входного порта (порта записи данных) указывается в поле редактирования *Write Width*. Введенное значение также автоматически отображается в поле *Read Width*, определяющем разрядность порта чтения данных. Пользователь не может изменять значение параметра *Read Width* при создании описаний элементов запоминающих устройств FIFO, в которых используется один сигнал синхронизации для портов записи и чтения данных. Количество ячеек в генерируемом элементе FIFO-памяти задается в поле редактирования *Write Depth*. Это же значение автоматически отображается в поле *Read Depth*. Выбираемые значения разрядности порта записи данных *Write Width* и количества ячеек *Write Depth* не должны выходить за границы допустимых диапазонов, которые указаны в строках *Valid Range*, расположенных справа от соответствующих полей редактирования.

Чтобы задействовать функцию коррекции ошибок (*Error Correction Checking, ECC*) в формируемых элементах, следует установить в состояние «Включено» индикатор *Enable ECC*, представленный во встроенной панели *Implementation Options* (рис. 143). Данный индикатор состояния доступен только, если для реализации генерируемого запоминающего устройства используются ресурсы блочной памяти ПЛИС *Block SelectRAM* или встроенные модули FIFO в кристаллах семейства Virtex-5.

В процессе подготовки описаний элементов FIFO-памяти, предназначенных для последующей реализации на базе блочной памяти *Block SelectRAM* или встроенных модулей FIFO в кристаллах семейств Virtex-4 и Virtex-5, для повышения производительности генерируемых запоминающих устройств в ряде случаев можно задействовать встроенные выходные регистры. Для этого нужно перевести в состояние «Включено» ин-

дикатор *Use Embedded Registers in Block RAM or FIFO (when possible)*, который также расположен во встроенной панели *Implementation Options*. Если данный индикатор находится в состоянии «Включено», то выходные регистры устанавливаются в цепях сигналов DOUT и VALID. При этом нужно учитывать, что применение выходных регистров вносит дополнительную задержку при выполнении операции чтения данных, значение которой равно одному периоду тактового сигнала. Следует обратить также внимание на то, что при формировании запоминающих устройств FIFO, использующих режим чтения с функцией предварительного просмотра данных FWFT, реализуемых на основе модулей FIFO в ПЛИС семейства Virtex-5, встроенные выходные регистры всегда автоматически включаются в состав создаваемого элемента. В этом случае данные регистры задействуются для реализации функции предварительного просмотра данных. На рис. 145 представлены временные диаграммы, поясняющие выполнение операции чтения данных в запоминающих устройствах FIFO с одним сигналом синхронизации входного и выходного портов, которые реализуются на основе ресурсов блочной памяти ПЛИС Block SelectRAM с применением встроенных выходных регистров.

Информация о значении суммарной задержки чтения данных в генерируемых запоминающих устройствах FIFO отображается в строке *Read Latency (From Rising Edge of Read Clock)* (рис. 143).

Третья диалоговая панель «мастера» настройки параметров ядра FIFO Generator версии v4.1, вид которой показан на рис. 146, предоставляет возможность выбора дополнительных выходов сигналов состояния и подтверждения, включаемых в состав интерфейса создаваемых элементов памяти, работающих по принципу «первым вошел — первым вышел». В этой же диалоговой панели определяется необходимость применения режима (и соответствующего входа) сброса и его тип (синхронный или асинхронный).

Для применения в составе генерируемых запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», дополнительных выходов сигналов статуса необходимо воспользоваться индикаторами состояния, которые находятся во встроенной панели *Optional Flags* (рис. 146). Чтобы добавить в состав интерфейса формируемого элемента FIFO-памяти выход сигнала, информирующего о приближении заполнения всех ячеек, следует установить в состоянии «Включено» индикатор *Almost Full Flag*. Активный уровень сигнала на этом выходе сообщает о том, что в элемент памяти может быть записано еще не более одного слова данных. Если в создаваемом элементе необходимо задействовать выход сигнала, предупреждающего о приближении исчезновения данных, доступных для считывания

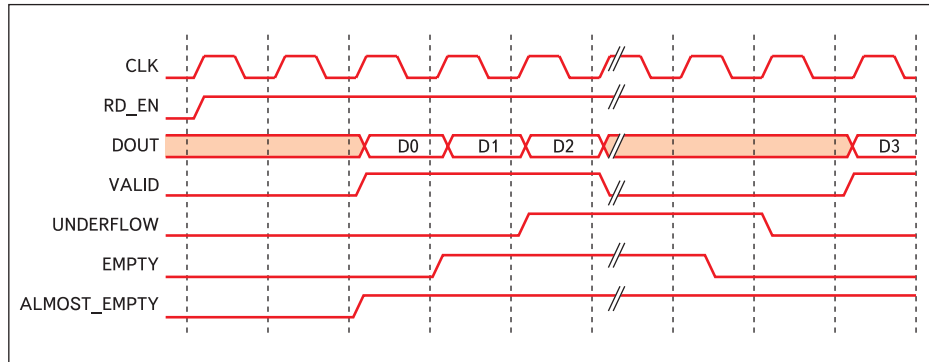


Рис. 145. Временные диаграммы, поясняющие выполнение операции чтения данных в элементах FIFO-памяти с одним сигналом синхронизации входного и выходного портов, которые реализуются на основе ресурсов блочной памяти ПЛИС Block SelectRAM с применением встроенных выходных регистров

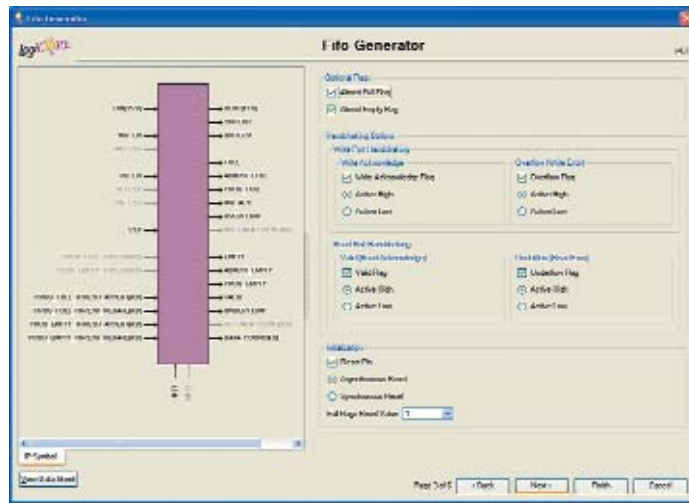


Рис. 146. Вид третьей диалоговой панели «мастера» настройки параметров генератора элементов памяти, работающих по принципу «первым вошел — первым вышел», FIFO Generator версии v4.1

из запоминающего устройства, то нужно перевести в состояние «Включено» индикатор *Almost Empty Flag*. Появление активного уровня сигнала на этом выходе информирует о том, что не более одного слова данных может быть еще считано из сформированного запоминающего устройства.

Выбор сигналов подтверждения, используемых в создаваемых элементах FIFO-памяти, и определение активных уровней этих сигналов осуществляется с помощью индикаторов состояния и кнопок, расположенных во встроенной панели *Handshaking Options* (рис. 146). Эта панель в свою очередь включает в себя две встроенные панели *Write Port Handshaking* и *Read Port Handshaking*. Для указания необходимых сигналов подтверждения, относящихся к порту записи данных, и назначения их активных уровней нужно использовать индикаторы состояния и кнопки, находящиеся во встроенной панели *Write Port Handshaking*. Чтобы добавить в состав интерфейса создаваемого запоминающего устройства FIFO выход сигнала Write Acknowledge, информирующего об успешном выполнении

записи данных в течение предыдущего периода тактового сигнала, следует установить в состояние «Включено» индикатор *Write Acknowledge Flag*. Активный уровень этого сигнала подтверждения записи выбирается с помощью двух кнопок с зависимой фиксацией, представленных во встроенной панели *Write Acknowledge*. Если в нажатом состоянии находится кнопка *Active High*, то активным уровнем сигнала на выходе Write Acknowledge будет считаться высокий логический уровень (уровень логической единицы). Если нажать кнопку *Active Low*, активным уровнем этого сигнала подтверждения будет являться низкий логический уровень (уровень логического нуля).

Для включения в состав интерфейса формируемого элемента FIFO-памяти выхода сигнала *Overflow*, сообщающего об ошибке в процессе выполнения операции записи во время предыдущего периода тактового сигнала из-за переполнения емкости запоминающего устройства, нужно установить в состоянии «Включено» индикатор *Overflow Flag*, который представлен во встроенной панели

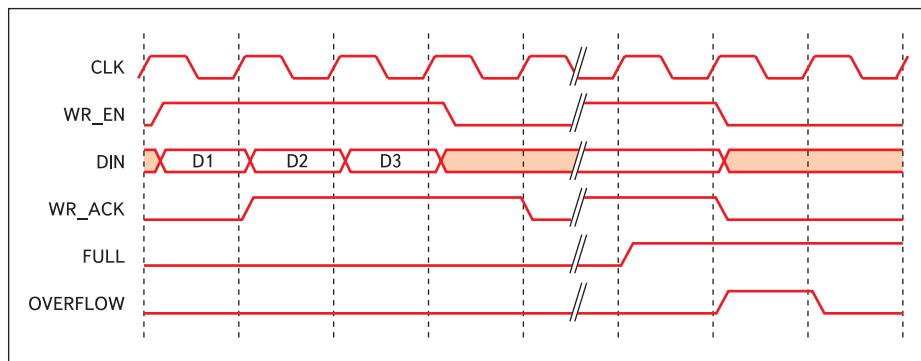


Рис. 147. Временные диаграммы, поясняющие формирование выходных сигналов подтверждения Write Acknowledge и Overflow в запоминающих устройствах FIFO, в которых используется один сигнал синхронизации для портов записи и чтения данных

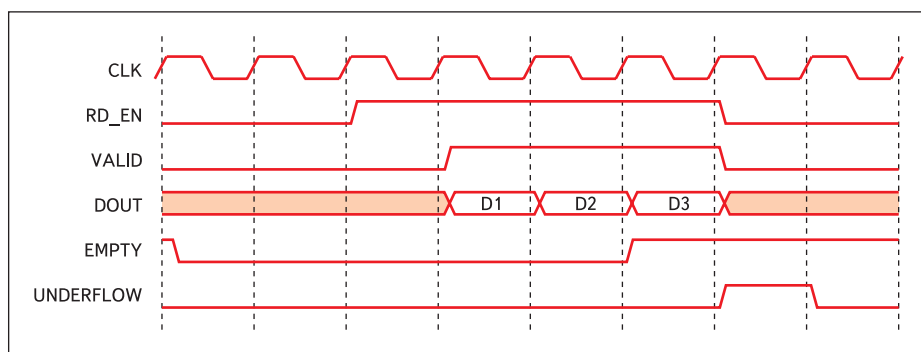


Рис. 148. Временные диаграммы, поясняющие формирование выходных сигналов подтверждения Valid и Underflow в запоминающих устройствах FIFO, в которых используется один сигнал синхронизации для портов записи и чтения данных

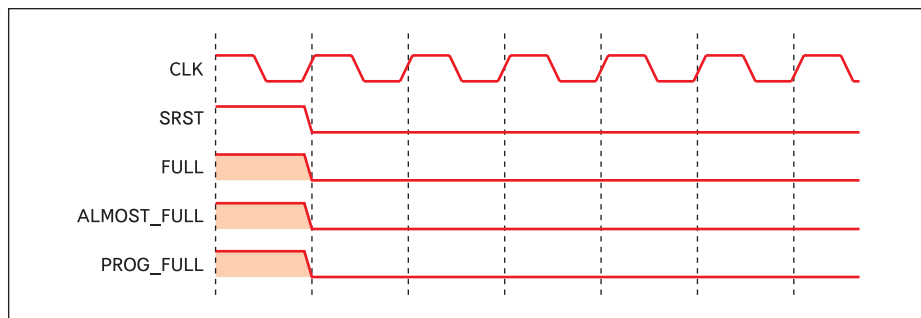


Рис. 149. Временные диаграммы, соответствующие режиму синхронного сброса запоминающих устройств FIFO с одним тактовым сигналом, реализуемых на базе ресурсов блочной или распределенной памяти ПЛИС

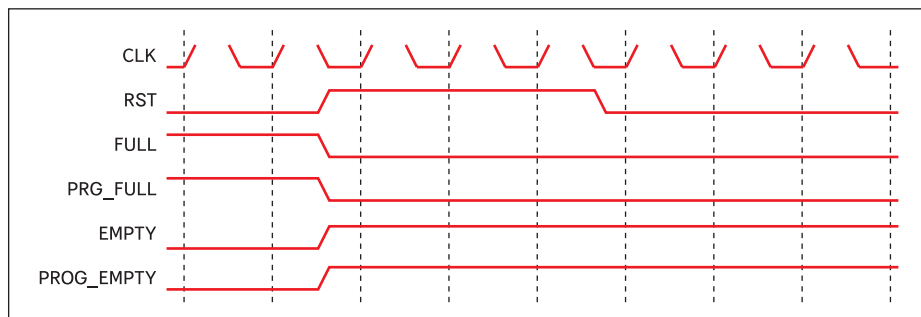


Рис. 150. Временные диаграммы, соответствующие режиму асинхронного сброса запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», с одним сигналом синхронизации портов записи и чтения данных, при реализации этих элементов на основе встроенных модулей FIFO в ПЛИС семейств Virtex-4 и Virtex-5

Overflow (Write Error). Две кнопки с зависимой фиксацией Active High и Active Low, находящиеся в этой же панели, позволяют выбрать активный уровень данного сигнала переполнения. На рис. 147 изображены временные диаграммы, поясняющие формирование выходных сигналов подтверждения *Write Acknowledge* и *Overflow* в запоминающих устройствах FIFO, в которых используется один сигнал синхронизации для портов записи и чтения данных.

Определение совокупности сигналов подтверждения, относящихся к порту чтения данных, и указание их активных уровней производится с помощью индикаторов состояния и кнопок с зависимой фиксацией, расположенных во встроенной панели *Read Port Handshaking* (рис. 146). Чтобы задействовать в генерируемом запоминающем устройстве FIFO выход сигнала Valid, информирующего о достоверности информации, представленной на выходной шине данных, необходимо перевести в состояние «Включено» индикатор *Valid Flag*, который находится во встроенной панели *Valid (Read Acknowledge)*. Выбор активного уровня этого сигнала подтверждения осуществляется с помощью двух кнопок с зависимой фиксацией *Active High* и *Active Low*.

Если в состав интерфейса создаваемого элемента FIFO-памяти необходимо добавить выход сигнала Underflow, извещающего об ошибке в процессе выполнения операции чтения в течение предыдущего периода тактового сигнала из-за отсутствия записанных данных, то следует установить в состояние «Включено» индикатор *Underflow Flag*, который расположен во встроенной панели *Underflow (Read Error)* (рис. 146). Активный уровень для этого сигнала указывается с помощью двух кнопок с зависимой фиксацией *Active High* и *Active Low*. Временные диаграммы, демонстрирующие последовательность формирования выходных сигналов подтверждения Valid и Underflow в запоминающих устройствах FIFO, в которых используется один сигнал синхронизации для портов записи и чтения данных, приведены на рис. 148.

В запоминающих устройствах, работающих по принципу «первым вошел — первым вышел», создаваемых с помощью параметризованного модуля FIFO Generator версии v4.1, может использоваться режим синхронного или асинхронного сброса. В режиме сброса производится обнуление всех внутренних счетчиков, регистров и содержимого ячеек памяти. Для реализации выбранного режима сброса в формируемом элементе должен быть предусмотрен соответствующий вход. Включение входа сброса в состав интерфейса создаваемого элемента FIFO-памяти и определение его типа осуществляется с помощью индикаторов состояния и кнопок, расположенных во встроенной панели *Initialization* (рис. 146). Чтобы добавить вход

сброса в состав интерфейса формируемого запоминающего устройства, следует установить в состояние «Включено» индикатор *Reset Pin*. Выбор необходимого варианта режима сброса производится с помощью двух кнопок с зависимой фиксацией — *Synchronous Reset* и *Asynchronous Reset*, которые также находятся во встроенной панели *Initialization*. Если в генерируемом элементе предполагается использование режима синхронного сброса, то следует нажать кнопку *Synchronous Reset*. Элементы FIFO-памяти, в которых для тактирования портов записи и чтения данных используется один сигнал синхронизации, поддерживают режим синхронного сброса только при реализации этих устройств на базе ресурсов блочной или распределенной памяти ПЛИС. Временные диаграммы, соответствующие режиму синхронного сброса запоминающих устройств FIFO с одним тактовым сигналом, реализуемых на базе ресурсов блочной или распределенной памяти ПЛИС, приведены на рис. 149.

Чтобы задействовать в создаваемом элементе FIFO-памяти режим асинхронного сброса, нужно нажать кнопку *Asynchronous Reset*. Значения выходных сигналов состояния генерируемого запоминающего устройства FIFO в режиме асинхронного сброса зависят от выбранного способа реализации этого элемента. На рис. 150 представлены временные диаграммы, соответствующие режиму асинхронного сброса запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», с одним сигналом синхронизации портов записи и чтения данных, при реализации этих элементов на основе встроенных модулей FIFO в ПЛИС семейств Virtex-4 и Virtex-5.

Если в генерируемом элементе FIFO-памяти, предназначенном для реализации на базе сдвиговых регистров, ресурсов блочной или распределенной памяти ПЛИС, выбирается режим асинхронного сброса, то разработчику предоставляется возможность выбора уровней сигналов состояния на выходах FULL, ALMOST_FULL и PROG_FULL в этом режиме. Для данной цели предназначено поле выбора *Full Flags Reset Value*, находящееся также во встроенной панели *Initialization*. Выпадающий список значений этого поля выбора содержит два варианта — 0 и 1, которые соответствуют низкому и высокому логическому уровню сигнала. На рис. 151 показаны временные диаграммы, соответствующие режиму асинхронного сброса запоминающих устройств FIFO с одним сигналом синхронизации портов записи и чтения данных, реализуемых на основе сдвиговых регистров, ресурсов блочной или распределенной памяти ПЛИС, при выборе нулевого значения для указанных выше сигналов состояния.

Временные диаграммы, соответствующие режиму асинхронного сброса FIFO-памяти с одним сигналом синхронизации портов за-

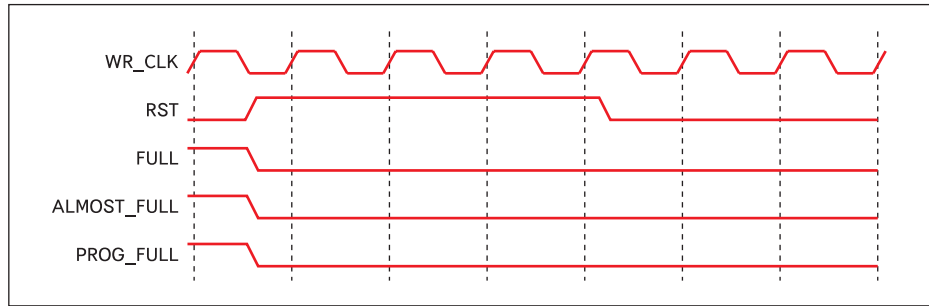


Рис. 151. Временные диаграммы, соответствующие режиму асинхронного сброса запоминающих устройств FIFO с одним сигналом синхронизации портов записи и чтения данных, реализуемых на основе сдвиговых регистров, ресурсов блочной или распределенной памяти ПЛИС, при выборе нулевого значения сигналов состояния FULL, ALMOST_FULL и PROG_FULL

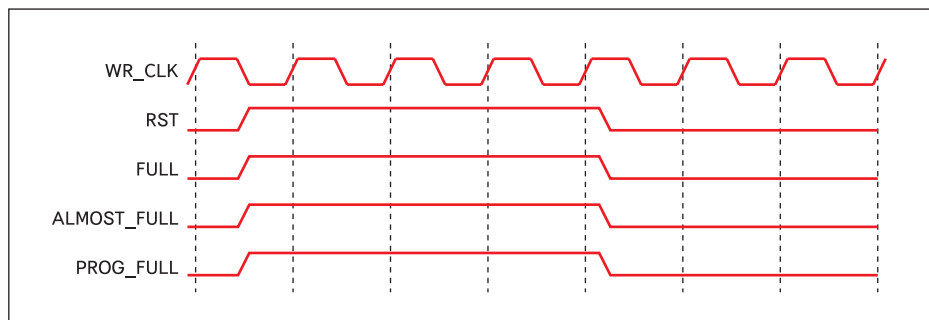


Рис. 152. Временные диаграммы, соответствующие режиму асинхронного сброса FIFO-памяти с одним сигналом синхронизации портов записи и чтения данных, при реализации этих элементов на базе сдвиговых регистров, ресурсов блочной или распределенной памяти ПЛИС, для случая выбора единичного значения сигналов состояния FULL, ALMOST_FULL и PROG_FULL

писи и чтения данных, при реализации этих элементов на базе сдвиговых регистров, ресурсов блочной или распределенной памяти ПЛИС, для случая выбора единичного значения сигналов состояния FULL, ALMOST_FULL и PROG_FULL изображены на рис. 152.

Четвертая диалоговая панель «мастера» настройки параметров ядра FIFO Generator версии v4.1 используется для выбора дополни-

тельных, программируемых пользователем, выходов сигналов состояния (флагов) и указания условий переключения этих сигналов. Вид данной диалоговой панели представлен на рис. 153.

Параметризованный модуль FIFO Generator версии v4.1 позволяет добавить в состав интерфейса генерируемого элемента дополнительные выходы сигналов состо-

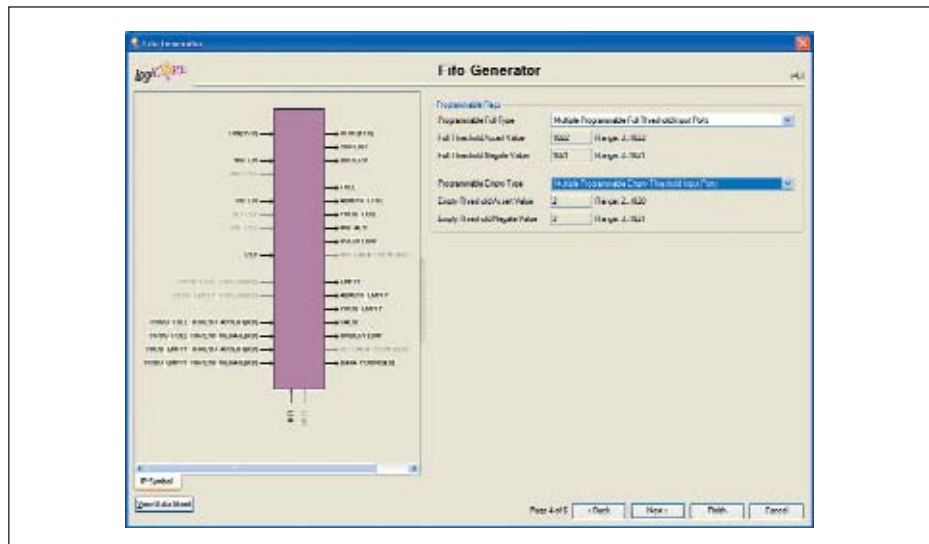


Рис. 153. Вид четвертой диалоговой панели «мастера» настройки параметров генератора элементов памяти, работающих по принципу «первым вошел — первым вышел», FIFO Generator версии v4.1

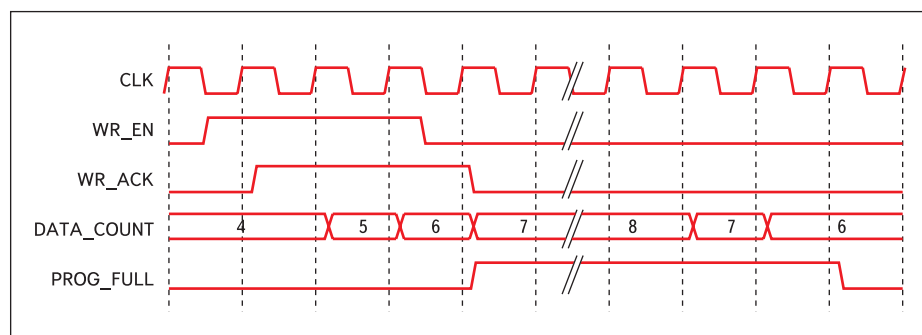


Рис. 154. Временные диаграммы, поясняющие процесс формирования дополнительного программируемого сигнала заполнения заданного числа ячеек PROG_FULL, условие переключения которого определяется одним параметром

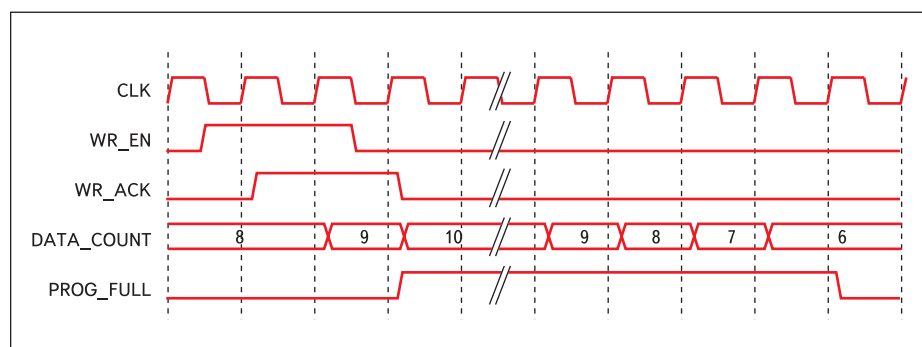


Рис. 155. Временные диаграммы, поясняющие процесс формирования дополнительного программируемого сигнала заполнения PROG_FULL, условия переключения которого определяются двумя параметрами

яния, информирующих о достижении верхней и нижней границ заданного диапазона количества слов данных, записанных в создаваемом запоминающем устройстве. Условия переключения этих сигналов определяет разработчик. Для этой цели предназначены поля выбора и редактирования, расположенные во встроенной панели *Programmable Flags* (рис. 153). Чтобы задействовать в формируемом элементе FIFO-памяти программируемый выход сигнала PROG_FULL, предупреждающего о заполнении указанного числа ячеек (записи заданного количества слов данных), следует воспользоваться полем выбора *Programmable Full Type*. Выпадающий список возможных значений этого параметра содержит пять вариантов: *No Programmable Full Threshold*, *Single Programmable Full Threshold Constant*, *Multiple Programmable Full Threshold Constants*, *Single Programmable Full Threshold Input Port*, *Multiple Programmable Full Threshold Input Ports*. По умолчанию в поле выбора *Programmable Full Type* указано значение *No Programmable Full Threshold*, при котором дополнительный программируемый выход сигнала заполнения PROG_FULL не используется.

В результате выбора варианта *Single Programmable Full Threshold Constant* в состав интерфейса формируемого запоминающего устройства FIFO включается выход сигнала, сообщающего о записи определенного числа слов данных (заполнении определенного числа ячеек), значение которого задается

в форме константы в поле редактирования *Full Threshold Assert Value*. На рис. 154 приведены временные диаграммы, поясняющие процесс формирования программируемого сигнала заполнения PROG_FULL, которые соответствуют этому варианту.

При выборе значения *Multiple Programmable Full Threshold Constants* в создаваемом элементе применяется выход сигнала PROG_FULL, условия переключения которого задаются в форме двух констант, определяемых с помощью полей редактирования *Full Threshold Assert Value* и *Full Threshold Negate Value*. В поле редактирования *Full Threshold Assert Value* указывается максимальное количество записанных слов данных, при достижении которого программируемый сигнал заполнения заданного числа ячеек устанавливается в активное состояние. Значение, задаваемое в поле редактирования *Full Threshold Negate Value*, определяет число записанных слов данных, при котором сигнал PROG_FULL переключается в сброшенное состояние. На рис. 155 представлены временные диаграммы, демонстрирующие процесс формирования программируемого сигнала заполнения заданного числа ячеек PROG_FULL, которые соответствуют варианту *Multiple Programmable Full Threshold Constants*.

Если в поле выбора *Programmable Full Type* указывается значение *Single Programmable Full Threshold Input Port*, то вместе с программируемым выходом сигнала заполнения PROG_FULL в состав интерфейса генериру-

емого запоминающего устройства добавляется входная шина PROG_FULL_THRESH. Двоичный код, представленный на этой шине, определяет количество записанных слов данных, при достижении которого производится переключение состояния сигнала PROG_FULL. Установка нового значения этого кода осуществляется в режиме сброса. Временные диаграммы, изображенные на рис. 154, справедливы и для рассматриваемого варианта, с учетом того, что пороговое значение количества слов данных, определяющее условие переключения данного сигнала, может изменяться в процессе функционирования разрабатываемого устройства.

Когда для параметра *Programmable Full Type* выбирается вариант *Multiple Programmable Full Threshold Input Ports*, в создаваемом элементе FIFO-памяти кроме программируемого выхода сигнала заполнения PROG_FULL появляются две входные шины PROG_FULL_THRESH_ASSERT и PROG_FULL_THRESH_NEGATE. Двоичный код, подаваемый на шину PROG_FULL_THRESH_ASSERT, определяет количество записанных слов данных, при котором осуществляется установка активного уровня сигнала на этом выходе. Значение двоичного кода на шине PROG_FULL_THRESH_NEGATE указывает количество записанных слов данных, при котором производится сброс сигнала на программируемом выходе заполнения PROG_FULL. Временные диаграммы, показанные на рис. 155, также соответствуют значению *Multiple Programmable Full Threshold Input Ports* параметра *Programmable Full Type* с учетом возможности изменения пороговых значений сброса и установки сигнала PROG_FULL в процессе работы проектируемого устройства.

Следует обратить внимание на то, что в элементах памяти, работающих по принципу «первым вошел — первым вышел», предназначенных для последующей реализации на базе встроенных модулей FIFO в ПЛИС семейств Virtex-4 и Virtex-5, можно использовать дополнительный выход сигнала заполнения PROG_FULL, для которого условие переключения задается только в форме константы. Таким образом, при подготовке описаний запоминающих устройств, реализуемых на основе встроенных модулей FIFO, список возможных значений для параметра *Programmable Full Type* содержит только два варианта — *No Programmable Full Threshold* и *Single Programmable Full Threshold Constant*.

Для включения в состав интерфейса генерируемого запоминающего устройства FIFO дополнительного программируемого выхода сигнала PROG_EMPTY, информирующего о достижении заданного минимально-пустимого числа записанных слов данных, необходимо воспользоваться полем выбора *Programmable Empty Type* (рис. 153). В выпадающем списке этого поля выбора представлено пять вариантов возможных значений:

No Programmable Empty Threshold, Single Programmable Empty Threshold Constant, Multiple Programmable Empty Threshold Constants, Single Programmable Empty Threshold Input Port, Multiple Programmable Empty Threshold Input Ports. Значение *No Programmable Empty Threshold*, предлагаемое по умолчанию, устанавливает конфигурацию элемента FIFO-памяти, в которой дополнительный программируемый выход сигнала состояния *PROG_EMPTY* не используется. Остальные четыре варианта позволяют задействовать выход данного сигнала, одновременно определяя условия его переключения.

Чтобы добавить в состав интерфейса формируемого элемента выход сигнала, сообщающего о достижении в процессе чтения (извлечения) информации определенного минимального числа записанных слов данных, значение которого задается в форме константы в поле редактирования *Empty Threshold Assert Value*, нужно в поле выбора *Programmable Empty Type* указать вариант *Single Programmable Empty Threshold Constant*. Если это значение должно изменяться в процессе функционирования разрабатываемого устройства, то следует выбрать вариант *Single Programmable Empty Threshold Input Port*. В этом случае в состав интерфейса также включается входная шина *PROG_EMPTY_THRESH*, на которую поступает двоичный код, определяющий минимально-допустимое число записанных слов данных, при котором производится переключение сигнала состояния *PROG_EMPTY*. Процесс формирования программируемого сигнала состояния *PROG_EMPTY*, условие переключения которого определяется одним параметром, демонстрируют временные диаграммы, представленные на рис. 156.

Когда необходимо задать различные условия (пороговые значения) для установки активного уровня и сброса дополнительного сигнала состояния *PROG_EMPTY*, следует выбрать вариант *Multiple Programmable Empty Threshold Constants*. В этом случае становятся доступными поля редактирования *Empty Threshold Assert Value* и *Empty Threshold Negate Value* (рис. 153). В поле редактирования *Empty Threshold Assert Value* определяется минимальное значение количества записанных слов данных, при достижении которого в процессе чтения (извлечения) информации сигнал *PROG_EMPTY* устанавливается в активное состояние. Значение, указываемое в поле редактирования *Empty Threshold Negate Value*, задает минимальное число записанных слов данных, при котором сигнал *PROG_EMPTY* переключается в сброшенное состояние. Если значения, определяющие условия установки активного уровня и сброса дополнительного сигнала состояния *PROG_EMPTY*, должны загружаться в процессе работы проектируемого устройства, то в поле выбора *Programmable Empty Type* нужно указать вариант *Multiple Programmable Empty Threshold*

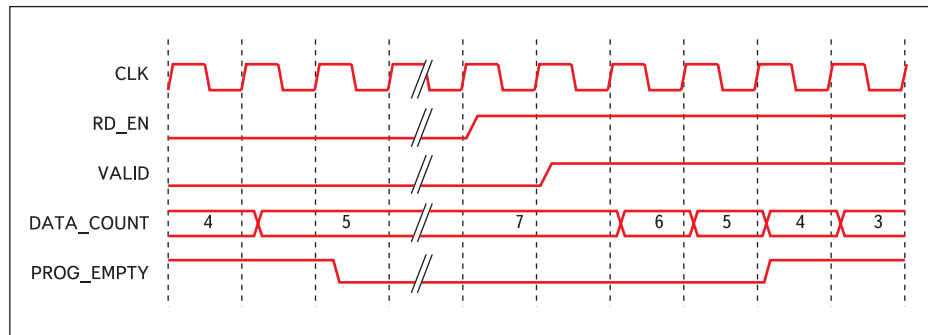


Рис. 156. Временные диаграммы, поясняющие процесс формирования дополнительного сигнала состояния *PROG_EMPTY*, условие переключения которого определяется одним параметром

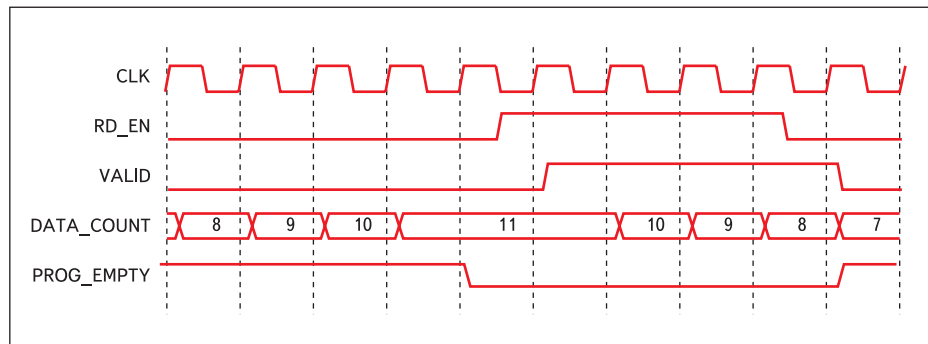


Рис. 157. Временные диаграммы, поясняющие процесс формирования дополнительного сигнала состояния *PROG_EMPTY* с различными условиями установки активного уровня и сброса

Input Ports. При этом в состав интерфейса генерируемого запоминающего устройства FIFO дополнительно включаются две входные шины *PROG_EMPTY_THRESH_ASSERT* и *PROG_EMPTY_THRESH_NEGATE*. Двоичный код, представленный на шине *PROG_EMPTY_THRESH_ASSERT*, определяет минимальное количество записанных слов данных, при котором в процессе выполнения операции чтения осуществляется установка активного уровня сигнала на выходе состояния *PROG_EMPTY*. Значение двоичного кода на шине *PROG_EMPTY_THRESH_NEGATE* задает минимальное число записанных слов данных, при котором производится сброс сигнала на программируемом выходе *PROG_EMPTY*. Процесс формирования программируемого сигнала состояния *PROG_EMPTY* с различными условиями установки активного уровня и сброса иллюстрируют временные диаграммы, приведенные на рис. 157.

В запоминающих устройствах, функционирующих по принципу «первым вошел — первым вышел», реализуемых на основе встроенных модулей FIFO в кристаллах семейств *Virtex-4* и *Virtex-5*, можно применять дополнительный программируемый выход сигнала состояния *PROG_EMPTY*, для которого условие переключения задается только в форме константы. Поэтому при создании описаний таких элементов FIFO-памяти в списке возможных значений параметра *Programmable Empty Type* представлено только два варианта — *No Programmable Empty*

Threshold и *Single Programmable Empty Threshold Constant*.

Пятая диалоговая панель «мастера» настройки параметров ядра FIFO Generator версии v4.1, вид которой показан на рис. 158, предоставляет возможность использования в создаваемых элементах выходов счетчика записанных данных и определения состояния информационных выходов в режиме сброса.

При подготовке описаний запоминающих устройств, функционирующих по принципу «первым вошел — первым вышел», для последующей реализации на базе сдвиговых регистров, ресурсов блочной или распределенной памяти кристалла в состав интерфейса формируемого элемента может быть включена выходная шина, на которой отображается текущее значение счетчика числа слов данных, записанных в память. Для этого нужно установить в состоянии «Включено» индикатор *Data Count (Synchronized With Clk)*, который находится во встроенной панели *Data Count Options*. При этом становится доступным поле редактирования *Data Count Width*, в котором указывается разрядность выходной шины счетчика записанных данных. По умолчанию в этом поле редактирования автоматически отображается максимально возможное значение разрядности, соответствующее информационной емкости генерируемого элемента. При необходимости можно указать меньшее значение, но при этом часть разрядов счетчика данных не будет отображаться на выходной шине.

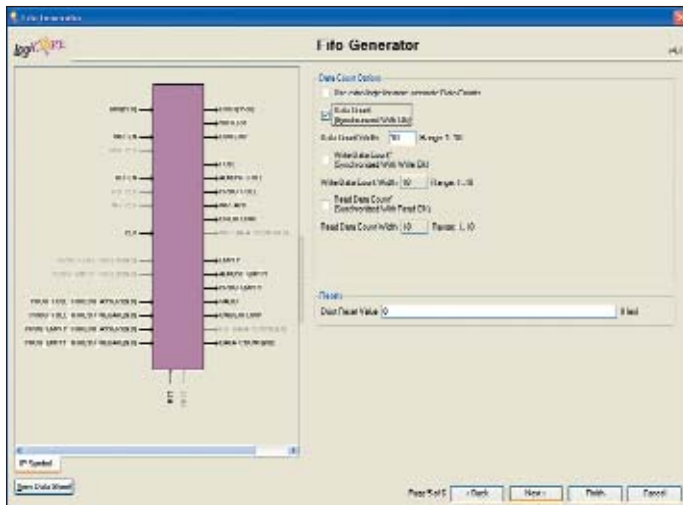


Рис. 158. Вид пятой диалоговой панели «мастера» настройки параметров генератора элементов памяти, работающих по принципу «первым вошел — первым вышел», FIFO Generator версии v4.1

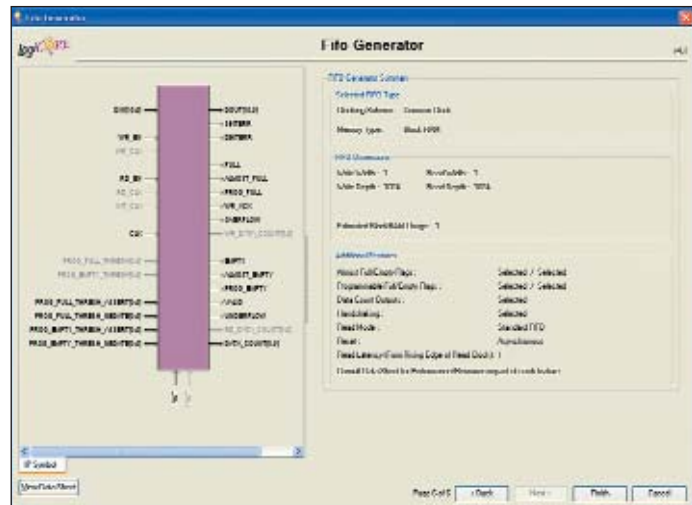


Рис. 159. Вид заключительной диалоговой панели «мастера» настройки параметров генератора элементов памяти, работающих по принципу «первым вошел — первым вышел», FIFO Generator версии v4.1

В процессе формирования описаний элементов FIFO-памяти, предназначенных для последующей реализации на основе модулей блочной памяти ПЛИС Block SelectRAM, ресурсов распределенной памяти кристалла или сдвиговых регистров, можно указать значения, отображаемые на выходной шине данных в режиме сброса. Для этой цели следует воспользоваться полем редактирования *Dout Reset Value*, которое расположено во встроенной панели *Resets*. По умолчанию в этом поле редактирования предлагается нулевое значение. Требуемое значение указывается в данном поле в виде шестнадцатеричного числа, количество разрядов которого соответствует разрядности считываемых данных.

В шестой, заключительной диалоговой панели «мастера» настройки параметров ядра FIFO Generator версии v4.1 отображается информация о выбранных опциях генерируемого запоминающего устройства, функционирующего по принципу «первым вошел — первым вышел». Вид этой диалоговой панели представлен на рис. 159.

Заключительная диалоговая панель *FIFO Generator Summary* содержит три встроенных информационных панели: *Selected FIFO Type*, *FIFO Dimensions* и *Additional Features*. В информационной панели *Selected FIFO Type* приводятся сведения о выбранном методе тактирования портов чтения и записи данных в формируемом элементе FIFO-памяти и ресурсах кристалла, используемых для реализации этого запоминающего устройства. Встроенная панель *FIFO Dimensions* содержит информацию об указанных параметрах организации портов записи и чтения данных в создаваемом запоминающем устройстве. В этой же встроенной панели, в строке *Estimated BlockRAM Usage*, приводится предварительная оценка количества модулей блочной памяти ПЛИС Block SelectRAM, задействованных в составе генерируемого за-

поминающего устройства FIFO. В информационной панели *Additional Features* отображаются сведения обо всех дополнительных выходах сигналов состояния и подтверждения, включаемых в состав интерфейса формируемого элемента FIFO-памяти. Здесь же представлена информация о выбранном режиме чтения и режиме сброса. Кроме того, в этой же встроенной панели, в строке *Read Latency (From Rising Edge of Read Clock)* приводится значение суммарной длительности задержки при выполнении операций чтения данных, выраженной в количестве периодов тактового сигнала. При необходимости, для изменения выбранных значений параметров конфигурации создаваемого запоминающего устройства следует воспользоваться кнопкой *Назад (Back)*, которая позволяет вернуться к предыдущим диалоговым панелям «мастера» настройки параметров ядра FIFO Generator версии v4.1.

Пример описания элемента запоминающего устройства FIFO с одним сигналом синхронизации портов записи и чтения данных, сформированного с помощью параметризованного модуля FIFO Generator версии v4.1 для последующей реализации на основе блочной памяти ПЛИС Block SelectRAM

В качестве примера описания запоминающего устройства FIFO с одним сигналом синхронизации портов записи и чтения данных, сформированного с помощью параметризованного модуля FIFO Generator версии v4.1 для последующей реализации на основе блочной памяти ПЛИС Block SelectRAM, в настоящем разделе приводится текст VHDL-описания элемента *fifo_generator_v4_1_common_clk_b_ram*. Данный элемент обладает ин-

формационной емкостью 16 кбит с организацией портов записи и чтения данных 1 К слов×16 разрядов. В сформированном запоминающем устройстве задействована функция коррекция ошибок ECC, а также встроенные выходные регистры в модулях блочной памяти Block SelectRAM.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
-- synthesis translate_off
Library XilinxCoreLib;
-- synthesis translate_on
ENTITY fifo_generator_v4_1_common_clk_b_ram IS
    port (
        clk: IN std_logic;
        din: IN std_logic_VECTOR(15 downto 0);
        prog_empty_thresh_assert: IN std_logic_VECTOR(9 downto 0);
        prog_empty_thresh_negate: IN std_logic_VECTOR(9 downto 0);
        prog_full_thresh_assert: IN std_logic_VECTOR(9 downto 0);
        prog_full_thresh_negate: IN std_logic_VECTOR(9 downto 0);
        rd_en: IN std_logic;
        rst: IN std_logic;
        wr_en: IN std_logic;
        almost_empty: OUT std_logic;
        almost_full: OUT std_logic;
        data_count: OUT std_logic_VECTOR(9 downto 0);
        dout: OUT std_logic_VECTOR(15 downto 0);
        empty: OUT std_logic;
        full: OUT std_logic;
        overflow: OUT std_logic;
        prog_empty: OUT std_logic;
        prog_full: OUT std_logic;
        valid: OUT std_logic;
        underflow: OUT std_logic;
        wr_ack: OUT std_logic;
        sbiterr: OUT std_logic;
        dbiterr: OUT std_logic
    );
END fifo_generator_v4_1_common_clk_b_ram;
--
ARCHITECTURE fifo_generator_v4_1_common_clk_b_ram_a OF
    fifo_generator_v4_1_common_clk_b_ram IS
    -- synthesis translate_off
    component wrapped_fifo_generator_v4_1_common_clk_b_ram
        port (
            clk: IN std_logic;
            din: IN std_logic_VECTOR(15 downto 0);
            prog_empty_thresh_assert: IN std_logic_VECTOR(9 downto 0);
            prog_empty_thresh_negate: IN std_logic_VECTOR(9 downto 0);
            prog_full_thresh_assert: IN std_logic_VECTOR(9 downto 0);
            prog_full_thresh_negate: IN std_logic_VECTOR(9 downto 0);
            rd_en: IN std_logic;
            rst: IN std_logic;
            wr_en: IN std_logic;
            almost_empty: OUT std_logic;
            almost_full: OUT std_logic;

```



```

data_count: OUT std_logic_VECTOR(9 downto 0);
dout: OUT std_logic_VECTOR(15 downto 0);
empty: OUT std_logic;
full: OUT std_logic;
overflow: OUT std_logic;
prog_empty: OUT std_logic;
prog_full: OUT std_logic;
valid: OUT std_logic;
underflow: OUT std_logic;
wr_ack: OUT std_logic;
sbiterr: OUT std_logic;
dbiterr: OUT std_logic
);
end component;
--
-- Configuration specification
for all : wrapped_fifo_generator_v4_1_common_clk_b_ram use
entity XilinxCoreLib.fifo_generator_v4_1 (behavioral)
generic map(
    c_has_int_clk => 0,
    c_rd_freq => 1,
    c_wr_response_latency => 1,
    c_has_srst => 0,
    c_has_rd_data_count => 0,
    c_din_width => 16,
    c_has_wr_data_count => 0,
    c_full_flags_rst_val => 1,
    c_implementation_type => 0,
    c_family => «virtex5»,
    c_use_embedded_reg => 1,
    c_has_wr_rst => 0,
    c_wr_freq => 1,
    c_underflow_low => 0,
    c_has_meminit_file => 0,
    c_has_overflow => 1,
    c_preload_latency => 2,
    c_dout_width => 16,
    c_rd_depth => 1024,
    c_default_value => «BlankString»,
    c_mif_file_name => «BlankString»,
    c_has_underflow => 1,
    c_has_rd_rst => 0,
    c_has_almost_full => 1,
    c_has_rst => 1,
    c_data_count_width => 10,
    c_has_wr_ack => 1,
    c_use_ecc => 1,
    c_wr_ack_low => 0,
    c_common_clock => 1,
    c_rd_pntr_width => 10,
    c_use_fwft_data_count => 0,
    c_has_almost_empty => 1,
    c_rd_data_count_width => 10,
    c_enable_rlocs => 0,
    c_wr_pntr_width => 10,
    c_overflow_low => 0,
    c_prog_empty_type => 4,
    c_optimization_mode => 0,
    c_wr_data_count_width => 10,
    c_preload_regs => 1,
    c_dout_rst_val => «0»,
    c_has_data_count => 1,
    c_prog_full_thresh_negate_val => 1021,
    c_wr_depth => 1024,
    c_prog_empty_thresh_negate_val => 3,
    c_prog_empty_thresh_assert_val => 2,
    c_has_valid => 1,
    c_init_wr_pntr_val => 0,
    c_prog_full_thresh_assert_val => 1022,
    c_use_fifo16_flags => 0,
    c_has_backup => 0,
    c_valid_low => 0,
    c_prim_fifo_type => «512x72»,
    c_count_type => 0,
    c_prog_full_type => 4,
    c_memory_type => 1
);
-- synthesis translate_on
BEGIN
-- synthesis translate_off
U0 : wrapped_fifo_generator_v4_1_common_clk_b_ram
port map (
    clk => clk,
    din => din,
    prog_empty_thresh_assert =>
prog_empty_thresh_assert,
    prog_empty_thresh_negate =>
prog_empty_thresh_negate,
    prog_full_thresh_assert => prog_full_thresh_assert,
    prog_full_thresh_negate => prog_full_thresh_negate,
    rd_en => rd_en,
    rst => rst,

```

```

wr_en => wr_en,
almost_empty => almost_empty,
almost_full => almost_full,
data_count => data_count,
dout => dout,
empty => empty,
full => full,
overflow => overflow,
prog_empty => prog_empty,
prog_full => prog_full,
valid => valid,
underflow => underflow,
wr_ack => wr_ack,
sbiterr => sbiterr,
dbiterr => dbiterr
);
-- synthesis translate_on
END fifo_generator_v4_1_common_clk_b_ram_a;

```

В состав интерфейса элемента *fifo_generator_v4_1_common_clk_b_ram* включены выходы сигналов, информирующих о том, что в запоминающее устройство может быть записано или извлечено из памяти не более одного слова данных, а также выходы 10-разрядного счетчика записанных данных и вход сигнала асинхронного сброса. Кроме того, в этом элементе задействованы дополнительные программируемые пользователем выходы сигналов состояния, условия переключения которых могут изменяться в процессе функционирования запоминающего устройства. Для каждого из этих сигналов условия переключения задаются в виде двух значений, представленных в виде двоичного кода на соответствующих входных шинах. Сгенерированный элемент памяти *fifo_generator_v4_1_common_clk_b_ram* полностью поддерживает совокупность сигналов подтверждения для портов записи и чтения данных, которые предусмотрены в параметризованном модуле FIFO Generator версии v4.1. В качестве активного уровня для этих сигналов установлен высокий логический уровень.

Чтобы использовать рассматриваемый элемент FIFO-памяти в качестве одного из компонентов в составе описания разрабатываемого устройства, следует добавить в раздел декларации VHDL-описания следующую совокупность выражений.

```

component fifo_generator_v4_1_common_clk_b_ram
port (
    clk: IN std_logic;
    din: IN std_logic_VECTOR(15 downto 0);
    prog_empty_thresh_assert: IN std_logic_VECTOR(9 downto 0);
    prog_empty_thresh_negate: IN std_logic_VECTOR(9 downto 0);
    prog_full_thresh_assert: IN std_logic_VECTOR(9 downto 0);
    prog_full_thresh_negate: IN std_logic_VECTOR(9 downto 0);
    rd_en: IN std_logic;
    rst: IN std_logic;
    wr_en: IN std_logic;
    almost_empty: OUT std_logic;
    almost_full: OUT std_logic;
    data_count: OUT std_logic_VECTOR(9 downto 0);
    dout: OUT std_logic_VECTOR(15 downto 0);
    empty: OUT std_logic;
    full: OUT std_logic;
    overflow: OUT std_logic;
    prog_empty: OUT std_logic;
    prog_full: OUT std_logic;
    valid: OUT std_logic;
    underflow: OUT std_logic;
    wr_ack: OUT std_logic;
    sbiterr: OUT std_logic;
    dbiterr: OUT std_logic
);

```

```

end component;
--
-- Synplicity black box declaration
attribute syn_black_box : boolean;
attribute syn_black_box of fifo_generator_v4_1_common_clk_b_ram:
component is true;

```

Для создания конкретного экземпляра компонента запоминающего устройства *fifo_generator_v4_1_common_clk_b_ram* в описании проектируемого устройства следует использовать оператор, шаблон которого имеет следующий вид:

```

<идентификатор_экземпляра_компонента_fifo_generator_v4_1_
common_clk_b_ram>: fifo_generator_v4_1_common_clk_b_ram
port map (
    clk => clk,
    din => din,
    prog_empty_thresh_assert => prog_empty_thresh_assert,
    prog_empty_thresh_negate => prog_empty_thresh_negate,
    prog_full_thresh_assert => prog_full_thresh_assert,
    prog_full_thresh_negate => prog_full_thresh_negate,
    rd_en => rd_en,
    rst => rst,
    wr_en => wr_en,
    almost_empty => almost_empty,
    almost_full => almost_full,
    data_count => data_count,
    dout => dout,
    empty => empty,
    full => full,
    overflow => overflow,
    prog_empty => prog_empty,
    prog_full => prog_full,
    valid => valid,
    underflow => underflow,
    wr_ack => wr_ack,
    sbiterr => sbiterr,
    dbiterr => dbiterr
);

```

Подробные сведения об объеме различных ресурсов ПЛИС (в том числе и необходимом количестве модулей блочной памяти кристалла), используемых для автономной реализации запоминающего устройства *fifo_generator_v4_1_common_clk_b_ram*, приведены в информационной панели, представленной на рис. 160.

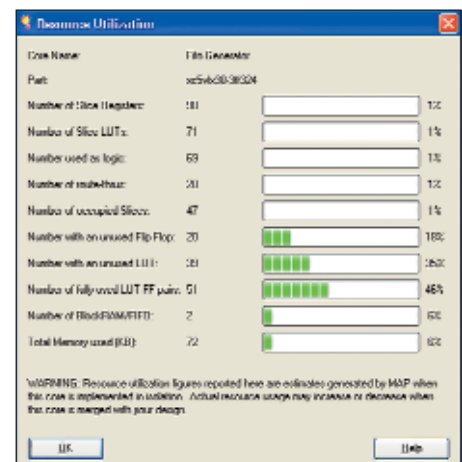


Рис. 160. Вид информационной панели, содержащей сведения о количестве различных ресурсов ПЛИС, используемых для реализации элемента FIFO-памяти *fifo_generator_v4_1_common_clk_b_ram*

Продолжение следует