

Окончание. Начало в № 1, 2'2004.

# Система команд микропроцессорного ядра MicroBlaze

Валерий Зотов

walerry@euro.ru

## Команды управления последовательностью выполнения операций в программе

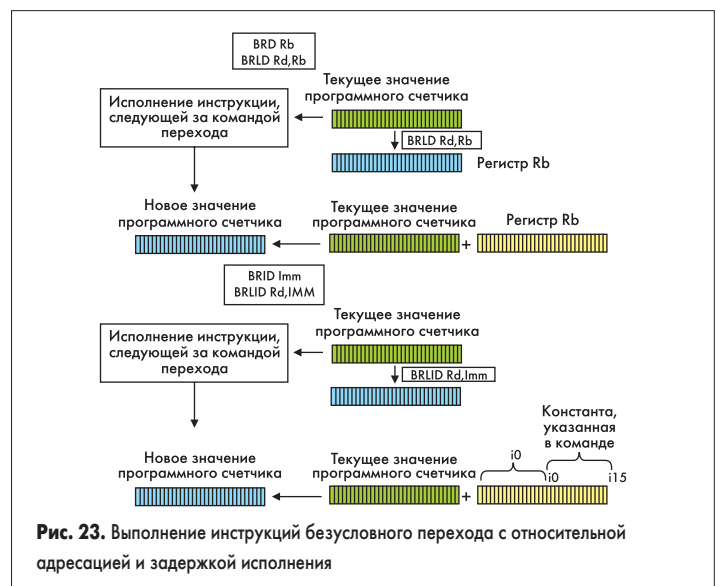
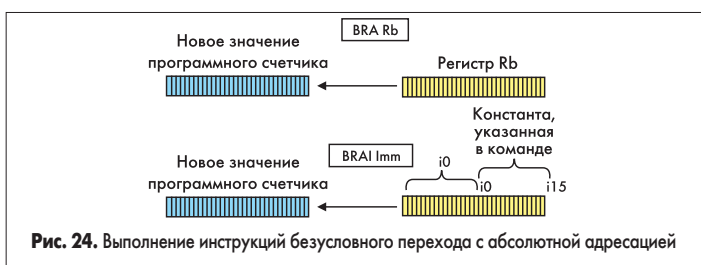
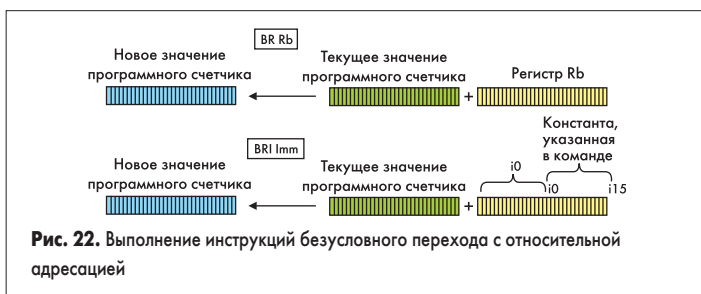
Группа инструкций, используемых для передачи управления в программе, включает в себя:

- команды безусловных переходов;
- команды условных переходов;
- команды возврата из подпрограмм и процедур обслуживания прерываний.

Система команд микропроцессорного ядра MicroBlaze содержит инструкции безусловных переходов, использующие как абсолютную, так и относительную адресацию. При абсолютной адресации исполнительный адрес, по которому осуществляется передача управления в программе, указывается в виде значения соответствующего параметра команды. В случае применения инструкций переходов, использующих относительную адресацию, исполнительный адрес вычисляется как сумма текущего значения содержимого программного счетчика и смещения, задаваемого в виде значения соответствующего параметра команды. Для определения значения адреса перехода (при абсолютной адресации) или смещения (при относительной адресации) может применяться регистровый косвенный или непосредственный способы. При регистровом косвенном способе значение адреса перехода или

смещения содержится в регистре общего назначения, номер которого указывается в качестве соответствующего параметра команды. Непосредственный способ позволяет задавать требуемые значения адреса или смещения в виде составной части (соответствующего поля) инструкции. Как правило, при этом значение адреса или смещения представляет собой результат знакового расширения 16-разрядной константы *Imm*, указанной непосредственно в инструкции, до 32-разрядного слова.

Большинство команд безусловных переходов представлено в двух вариантах: с немедленной передачей управления и с задержкой исполнения. При выполнении инструкции с немедленной передачей управления загрузка нового значения (адреса перехода) в программный счетчик осуществляется сразу же после завершения обработки этой команды. Инструкции безусловных переходов с задержкой исполнения применяются с целью сокращения временных потерь, вызванных необходимостью перезагрузки конвейера после их выполнения. При использовании этих команд фактическая передача управления происходит только после завершения исполнения следующей инструкции, находящейся в фазе дешифрации. В конце мнемонического идентификатора таких инструкций присутствует символ D (Delay), указывающий на то, что исполнение этих команд осуществляется с дополнительной задержкой.



Микропроцессорное ядро MicroBlaze поддерживает четырнадцать разновидностей команд безусловной передачи управления в программе, форматы которых приведены в таблице 17.

В инструкциях безусловного перехода *BR Rb* и *BRI Imm* используется относительная адресация с регистровым косвенным и непосредственным указанием значения смещения соответственно. Команда *BR* передает управление по адресу, равному сумме значений текущего состояния программного счетчика и содержимого регистра общего назначения, номер которого указывается в качестве единственного параметра инструкции. При использовании команды *BRI* адрес перехода представляет собой сумму текущего значения программного счетчика и результата знакового расширения 16-разрядной константы *Imm*, указанной непосредственно в инструкции, до 32-разрядного слова. Рисунок 22 демонстрирует выполнение команд безусловного перехода с относительной адресацией.

Команды *BRD Rb* и *BRID Imm* также предназначены для осуществления безусловного

перехода с относительной адресацией, но передача управления осуществляется только после завершения обработки следующей (декодируемой) инструкции. Команды *BRLD Rd,Rb* и *BRLID Rd,Imm* выполняют те же операции, что и предыдущая пара инструкций, сохраняя предварительно текущее состояние программного счетчика в регистре общего назначения. Номер регистра, используемого для записи текущего состояния программного счетчика, указывается в качестве значения первого параметра этих команд. Рисунок 23 поясняет выполнение команд безусловного перехода с относительной адресацией и задержкой исполнения.

Инструкции *BRA Rb* и *BRAI Imm* реализуют функцию безусловной передачи управления в программе с использованием абсолютной адресации, с регистровым косвенным и непосредственным указанием значения адреса перехода соответственно. В команде *BRA* значение адреса перехода определяется содержимым регистра общего назначения, номер которого указывается в качестве единственного параметра инструкции. Команда

*BRAI* передает управление по адресу, который представляет собой результат знакового расширения 16-разрядной константы *Imm*, указанной непосредственно в инструкции, до 32-разрядного слова. Рис. 24 демонстрирует выполнение команд безусловного перехода с абсолютной адресацией.

Команды *BRAD Rb* и *BRAID Imm* позволяют осуществлять безусловный переход с абсолютной адресацией и задержкой исполнения, которая необходима для завершения обработки следующей инструкции (находящейся в стадии декодирования). Единственным отличием команд *BRALD Rd,Rb* и *BRALID Rd,Imm* от предыдущей пары инструкций является сохранение текущего состояния программного счетчика в регистре общего назначения, номер которого указывается в качестве значения первого параметра этих команд. На рис. 25 показана последовательность операций при выполнении команд безусловного перехода с абсолютной адресацией и задержкой исполнения.

Инструкции *BRK Rd,Rb* и *BRKI Rd,Imm* предназначены для реализации в программе

Таблица 17. Форматы команд безусловной передачи управления в программе

Поле кода операции	Поле номера первого регистра	Поле типа перехода	Поле номера второго регистра	Нулевые разряды	Мнемоника	Выполняемая операция
1 0 0 1 1 0	0 0 0 0 0	0 0 0 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BR Rb</i>	Безусловный переход с относительной адресацией и регистровым косвенным указанием смещения (адрес перехода формируется путем суммирования текущего значения программного счетчика и содержимого регистра Rb)
1 0 0 1 1 0	0 0 0 0 0	1 0 0 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BRD Rb</i>	Безусловный переход с относительной адресацией, регистровым косвенным указанием смещения и задержкой исполнения (до завершения обработки декодируемой инструкции)
1 0 0 1 1 0	d d d d d	1 0 1 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BRLD Rd,Rb</i>	Безусловный переход с относительной адресацией, регистровым косвенным указанием смещения, задержкой исполнения и сохранением текущего состояния программного счетчика в регистре Rd
1 0 0 1 1 0	0 0 0 0 0	0 1 0 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BRA Rb</i>	Безусловный переход с абсолютной регистровой косвенной адресацией (адрес перехода указывается в форме значения содержимого регистра Rb)
1 0 0 1 1 0	0 0 0 0 0	1 1 0 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BRAD Rb</i>	Безусловный переход с абсолютной регистровой косвенной адресацией и задержкой исполнения
1 0 0 1 1 0	d d d d d	1 1 1 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BRALD Rd,Rb</i>	Безусловный переход с абсолютной регистровой косвенной адресацией, задержкой исполнения и сохранением текущего состояния программного счетчика в регистре Rd
1 0 0 1 1 0	d d d d d	0 1 1 0 0	b b b b b	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<i>BRK Rd,Rb</i>	Переход в режим Break с передачей управления по адресу, указанному в виде значения содержимого регистра Rb
0 1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30 31		Номер разряда микрокоманды
Поле кода операции	Поле номера первого регистра	Поле значения константы	Поле значения константы	Мнемоника	Выполняемая операция	
1 0 1 1 1 0	0 0 0 0 0	0 0 0 0 0	i i	<i>BRI Imm</i>	Безусловный переход с относительной адресацией и непосредственным определением смещения (адрес перехода формируется путем суммирования текущего значения программного счетчика и знакового расширения константы Imm)	
1 0 1 1 1 0	0 0 0 0 0	1 0 0 0 0	i i	<i>BRID Imm</i>	Безусловный переход с относительной адресацией, непосредственным определением смещения и задержкой исполнения	
1 0 1 1 1 0	d d d d d	1 0 1 0 0	i i	<i>BRLID Rd,Imm</i>	Безусловный переход с относительной адресацией, непосредственным определением смещения, задержкой исполнения и сохранением текущего состояния программного счетчика в регистре Rd	
1 0 1 1 1 0	0 0 0 0 0	0 1 0 0 0	i i	<i>BRAI Imm</i>	Безусловный переход с непосредственной абсолютной адресацией (адрес перехода указывается в форме знакового расширения константы Imm)	
1 0 1 1 1 0	0 0 0 0 0	1 1 0 0 0	i i	<i>BRAID Imm</i>	Безусловный переход с непосредственной абсолютной адресацией и задержкой исполнения	
1 0 1 1 1 0	d d d d d	1 1 1 0 0	i i	<i>BRALID Rd,Imm</i>	Безусловный переход с непосредственной абсолютной адресацией, задержкой исполнения и сохранением текущего состояния программного счетчика в регистре Rd	
1 0 1 1 1 0	d d d d d	0 1 1 0 0	i i	<i>BRKI Rd,Imm</i>	Переход в режим Break с передачей управления по непосредственно указываемому адресу (адрес перехода указывается в форме знакового расширения константы Imm)	
0 1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30 31		Номер разряда микрокоманды

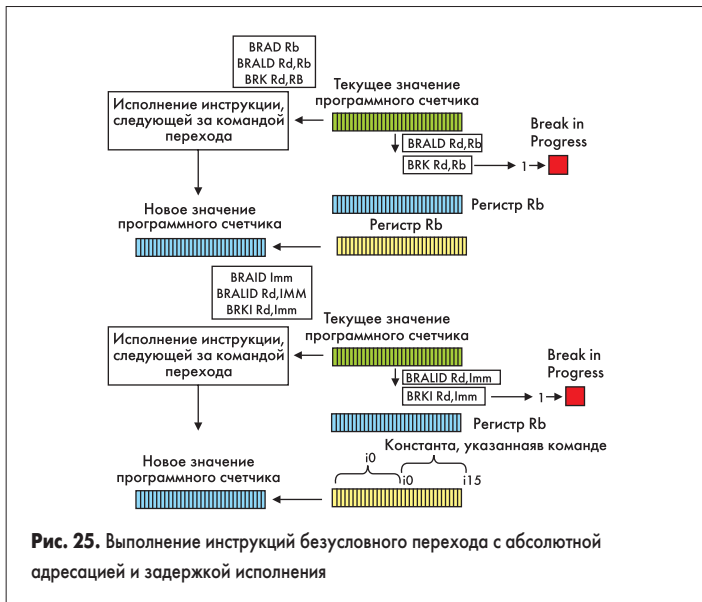


Рис. 25. Выполнение инструкций безусловного перехода с абсолютной адресацией и задержкой исполнения

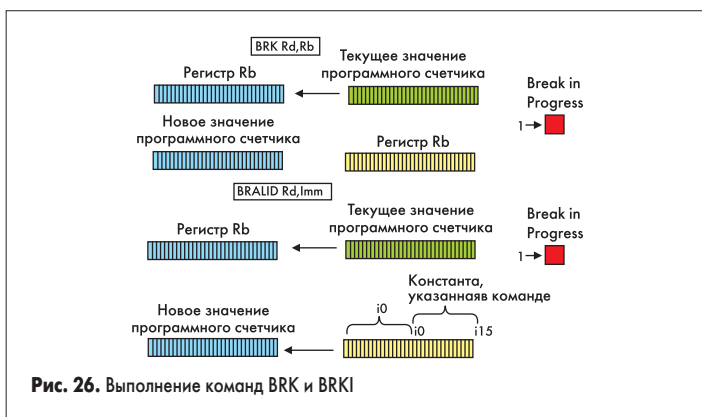


Рис. 26. Выполнение команд BRK и BRKI

режима Break с передачей управления по указанному адресу. В этих командах применяется абсолютная адресация. Первым параметром в данных инструкциях является номер одного из регистров общего назначения, который используется для записи значения текущего состояния программного счетчика. Содержимое этого регистра впоследствии может использоваться в процессе вычисления адреса возврата при выполнении соответствующей инструкции. В качестве значения второго параметра команды BRK также указывается номер регистра, содержимое которого определяет адрес передачи управления. В инструкции BRKI адрес перехода представляет собой результат знакового расширения 16-разрядной константы Imm, указанной непосредственно в инструкции, до 32-разрядного слова. При исполнении рассмотренных команд флаг Break in Progress Flag (28-й разряд регистра статуса) переключается в установленное состояние. Процесс выполнения команд BRK и BRKI иллюстрирует рисунок 26.

При условных переходах загрузка в программный счетчик нового адреса, по которому передается управление в программе, производится только при выполнении соответствующего условия. Если заданное условие не выполнено, то программный счетчик продолжает работу в инкрементном режиме. Проверка выполнения соответствующего условия заключается в сопоставлении с нулем значения содержимого одного из регистров общего назначения, номер которого указывается в ка-

честве первого параметра инструкций условных переходов. Система команд микропроцессорного ядра MicroBlaze содержит инструкции, позволяющие анализировать выполнение одного из шести типов условий:

- значение содержимого указанного регистра равно нулю;
- значение содержимого указанного регистра не равно нулю;
- значение содержимого указанного регистра меньше нуля;
- значение содержимого указанного регистра меньше или равно нулю;
- значение содержимого указанного регистра больше нуля;
- значение содержимого указанного регистра больше или равно нулю.

В отличие от команд безусловного перехода в инструкциях условных переходов всегда используется относительная адресация с регистровым смещением и непосредственным указанием значения смещения. Все команды условной передачи управления представлены в двух вариантах: с немедленным исполнением и с задержкой до завершения обработки следующей (декодируемой) инструкции. Форматы команд условных переходов представлены в таблице 18.

Инструкции BEQ Ra,Rb и BEQI Ra,Imm инициируют передачу управления по заданному адресу при условии, что значение содержимого регистра общего назначения с номером a равно нулю. Адрес перехода формируется путем суммирования текущего

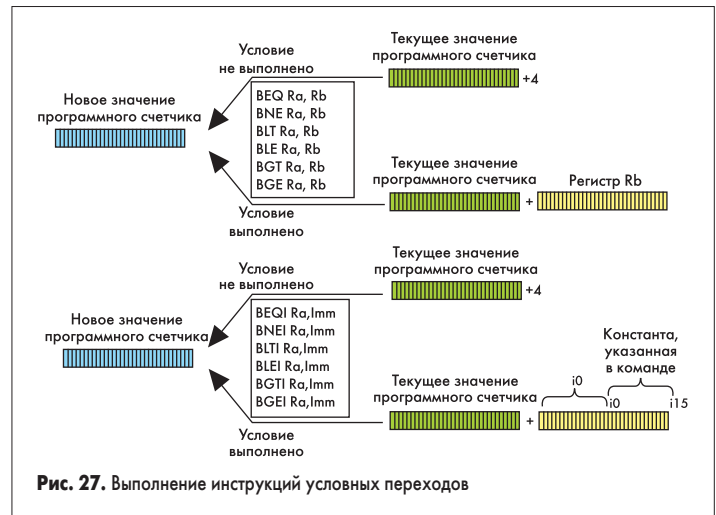


Рис. 27. Выполнение инструкций условных переходов

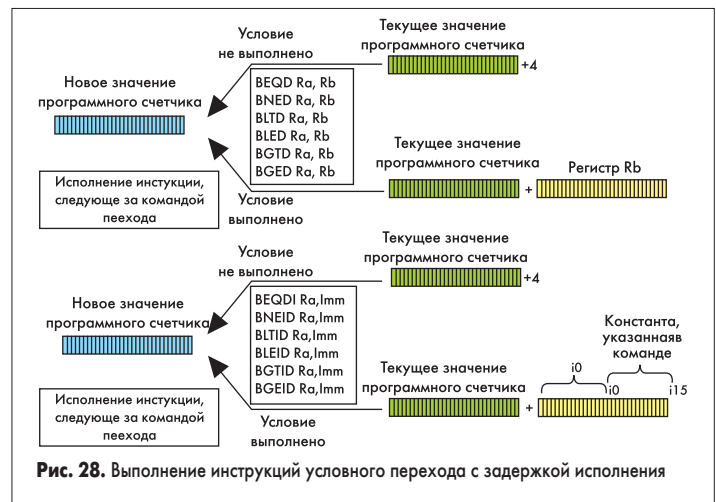


Рис. 28. Выполнение инструкций условного перехода с задержкой исполнения

значения программного счетчика и смещения, которое указывается в виде содержимого регистра Rb (в команде BEQ) или результата знакового расширения 16-разрядной константы Imm до 32-разрядного слова (в команде BEQI). При исполнении команд BNE Ra,Rb и BNEI Ra,Imm управление передается инструкции с указанным адресом только в случае, если значение содержимого регистра общего назначения с номером a не равно нулю. Инструкции BLT Ra,Rb и BLTI Ra,Imm загружают новое значение адреса в программный счетчик при условии, что значение содержимого регистра Ra меньше нуля. С помощью команд BLE Ra,Rb и BLEI Ra,Imm осуществляется переход к выполнению инструкции с указанным адресом, если содержимое регистра Ra меньше или равно нулю. Инструкции BGT Ra,Rb и BGTI Ra,Imm предназначены для передачи управления в программе по новому адресу при условии, что значение содержимого регистра Ra больше нуля. Команды BGE Ra,Rb и BGEI Ra,Imm позволяют выполнить переход к инструкции с требуемым адресом, если содержимое регистра Ra больше или равно нулю. Рисунок 27 поясняет выполнение команд условных переходов.

Инструкции BEQD Ra,Rb, BNED Ra,Rb, BLTD Ra,Rb, BLED Ra,Rb, BGTD Ra,Rb, BGED Ra,Rb, BEQID Ra,Imm, BNEID Ra,Imm, BLTID Ra,Imm, BLEID Ra,Imm, BGTID Ra,Imm, BGEID Ra,Imm выполняют те же функции, что и команды BEQ, BNE, BLT, BLE, BGT, BGE, BEQI, BNEI, BLTI, BLEI, BGTI, BGEI, рас-

Таблица 18. Форматы команд условных переходов

Поле кода операции					Поле типа условия перехода					Поле номера первого регистра					Поле номера второго регистра					Нулевые разряды										Мнемоника	Выполняемая операция										
1	0	0	1	1	1	0	0	0	0	0	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BEQ Ra,Rb	Переход при условии, что значение содержимого регистра Ra равно нулю, с регистровой косвенной адресацией (адрес перехода формируется путем суммирования текущего значения программного счетчика и содержимого регистра Rb)
1	0	0	1	1	1	0	0	0	0	1	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BNE Ra,Rb	Переход при условии, что значение содержимого регистра Ra не равно нулю, с регистровой косвенной адресацией	
1	0	0	1	1	1	0	0	0	1	0	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BLT Ra,Rb	Переход при условии, что значение содержимого регистра Ra меньше нуля, с регистровой косвенной адресацией
1	0	0	1	1	1	0	0	0	1	1	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BLE Ra,Rb	Переход при условии, что значение содержимого регистра Ra меньше или равно нулю, с регистровой косвенной адресацией
1	0	0	1	1	1	0	0	1	0	0	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BGT Ra,Rb	Переход при условии, что значение содержимого регистра Ra больше нуля, с регистровой косвенной адресацией
1	0	0	1	1	1	0	0	1	0	1	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BGE Ra,Rb	Переход при условии, что значение содержимого регистра Ra больше или равно нулю, с регистровой косвенной адресацией
1	0	0	1	1	1	1	0	0	0	0	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BEQD Ra,Rb	Переход при условии, что значение содержимого регистра Ra равно нулю, с регистровой косвенной адресацией и задержкой исполнения (до завершения обработки декодируемой инструкции)
1	0	0	1	1	1	1	0	0	0	1	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BNED Ra,Rb	Переход при условии, что значение содержимого регистра Ra не равно нулю, с регистровой косвенной адресацией и задержкой исполнения
1	0	0	1	1	1	1	0	0	1	0	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BLTD Ra,Rb	Переход при условии, что значение содержимого регистра Ra меньше нуля, с регистровой косвенной адресацией и задержкой исполнения
1	0	0	1	1	1	1	0	0	1	1	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BLEID Ra,Rb	Переход при условии, что значение содержимого регистра Ra меньше или равно нулю, с регистровой косвенной адресацией и задержкой исполнения
1	0	0	1	1	1	1	0	1	0	0	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BGTD Ra,Rb	Переход при условии, что значение содержимого регистра Ra больше нуля, с регистровой косвенной адресацией и задержкой исполнения
1	0	0	1	1	1	1	0	1	0	1	a	a	a	a	a	b	b	b	b	b	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BGED Ra,Rb	Переход при условии, что значение содержимого регистра Ra больше или равно нулю, с регистровой косвенной адресацией и задержкой исполнения
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Номер разряда микрокоманды									
Поле кода операции					Поле типа условия перехода					Поле номера первого регистра					Поле значения константы										Мнемоника	Выполняемая операция															
1	0	1	1	1	1	0	0	0	0	0	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BEQI Ra,Imm	Переход при условии, что значение содержимого регистра Ra равно нулю, с непосредственной адресацией (адрес перехода формируется путем суммирования текущего значения программного счетчика и знакового расширения константы Imm)	
1	0	1	1	1	1	0	0	0	0	1	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BNEI Ra,Imm	Переход при условии, что значение содержимого регистра Ra не равно нулю, с непосредственной адресацией	
1	0	1	1	1	1	0	0	0	1	0	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BLTI Ra,Imm	Переход при условии, что значение содержимого регистра Ra меньше нуля, с непосредственной адресацией	
1	0	1	1	1	1	0	0	0	1	1	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BLEI Ra,Imm	Переход при условии, что значение содержимого регистра Ra меньше или равно нулю, с непосредственной адресацией
1	0	1	1	1	1	0	0	1	0	0	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BGTI Ra,Imm	Переход при условии, что значение содержимого регистра Ra больше нуля, с непосредственной адресацией
1	0	1	1	1	1	0	0	1	0	1	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BGEI Ra,Imm	Переход при условии, что значение содержимого регистра Ra больше или равно нулю, с непосредственной адресацией
1	0	1	1	1	1	1	0	0	0	0	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BEQID Ra,Imm	Переход при условии, что значение содержимого регистра Ra равно нулю, с непосредственной адресацией и задержкой исполнения
1	0	1	1	1	1	1	0	0	0	1	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BNEID Ra,Imm	Переход при условии, что значение содержимого регистра Ra не равно нулю, с непосредственной адресацией и задержкой исполнения
1	0	1	1	1	1	1	0	0	1	0	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BLTID Ra,Imm	Переход при условии, что значение содержимого регистра Ra меньше нуля, с непосредственной адресацией и задержкой исполнения
1	0	1	1	1	1	1	0	0	1	1	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BLEID Ra,Imm	Переход при условии, что значение содержимого регистра Ra меньше или равно нулю, с непосредственной адресацией и задержкой исполнения
1	0	1	1	1	1	1	0	1	0	0	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BGTID Ra,Imm	Переход при условии, что значение содержимого регистра Ra больше нуля, с непосредственной адресацией и задержкой исполнения
1	0	1	1	1	1	1	0	1	0	1	a	a	a	a	a	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	BGEID Ra,Imm	Переход при условии, что значение содержимого регистра Ra больше или равно нулю, с непосредственной адресацией и задержкой исполнения
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Номер разряда микрокоманды									

смотренные выше. Отличие заключается в том, что передача управления производится при выполнении соответствующего условия только после завершения обработки следующей инструкции, находящейся в стадии дешифрации. Процесс выполнения команд

условного перехода с задержкой исполнения показан на рисунке 28.

В системе команд микропроцессорного ядра MicroBlaze представлено три инструкции возврата управления в программе:

- команда возврата из подпрограмм;

- команда возврата из процедур обслуживания прерываний;
- команда выхода из режима Break.

Все команды этой подгруппы предусматривают задержку передачи управления до момента завершения исполнения следую-





нала интерфейса FSL в регистр общего назначения с номером  $d$ , используя режим блокировки процессора. Инструкция  $nGET Rd,FSLn$  предназначена для чтения информационных данных из  $n$ -го канала интерфейса FSL в регистр  $Rd$  без блокировки. Считывание управляющих данных из  $n$ -го канала интерфейса FSL в регистр с номером  $d$  в режиме блокировки осуществляется с помощью команды  $cGET Rd,FSLn$ . Для чтения управляющих данных из  $n$ -го канала интерфейса FSL в регистр  $Rd$  без блокировки используется инструкция  $ncGET Rd,FSLn$ . Если при выполнении операций чтения значение контрольного бита не соответствует ожидаемому типу данных, то флаг FSL Error Flag (27-й разряд регистра статуса MSR) устанавливается в состояние, соответствующее высокому логическому уровню. При выполнении команд чтения без блокировки для достоверных данных флаг переноса Carry Flag (двадцать девятый разряд регистра статуса) сбрасывается в состояние низкого логического уровня. В случае чтения неверных данных флаг переноса устанавливается в состояние высокого логического уровня. Рисунок 30 поясняет выполнение инструкций чтения данных из каналов интерфейса FSL

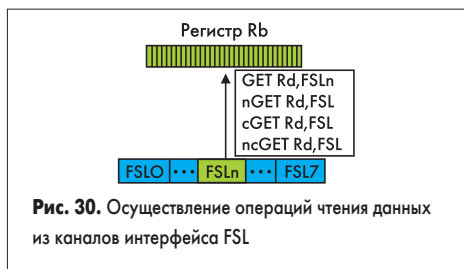


Рис. 30. Осуществление операций чтения данных из каналов интерфейса FSL

Для передачи данных различного типа из регистра общего назначения в выходной канал интерфейса FSL предусмотрены четыре варианта инструкций, форматы которых приведены в таблице 21.

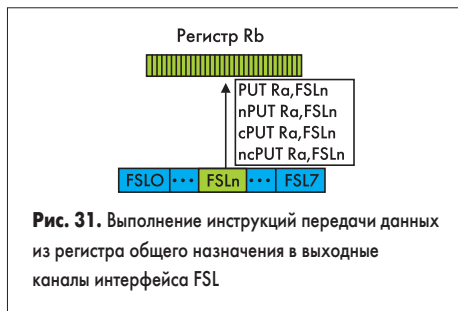


Рис. 31. Выполнение инструкций передачи данных из регистра общего назначения в выходные каналы интерфейса FSL

Передача информационных данных из регистра общего назначения с номером  $a$  в  $n$ -й канал интерфейса FSL в режиме блокировки осуществляется с помощью команды  $PUT Ra,FSLn$ . Для записи информационных данных из регистра  $Ra$  в  $n$ -й канал интерфейса FSL без блокировки предназначена команда  $nPUT Ra,FSLn$ . Команда  $cPUT Ra,FSLn$  выполняет операцию передачи управляющих данных из регистра  $Ra$  в  $n$ -й канал интерфейса FSL, используя режим блокировки процессора. Инструкция  $ncPUT Ra,FSLn$  осуществляет вывод управляющих данных из регистра  $Ra$  в  $n$ -й канал интерфейса FSL без блокировки. При успешном выполнении команд передачи данных из регистра общего назначения в выходной канал интерфейса FSL флаг переноса Carry Flag сбрасывается в состояние низкого логического уровня. В противном случае флаг переноса устанавливается в состояние высокого логического уровня. Рисунок 31 поясняет выполнение команд передачи данных различного типа из регистра общего назначения в выходной канал интерфейса FSL.

Группа специальных команд

Инструкции, входящие в данную группу, предназначены для копирования содержимого регистров специального назначения в регистры общего назначения и наоборот.

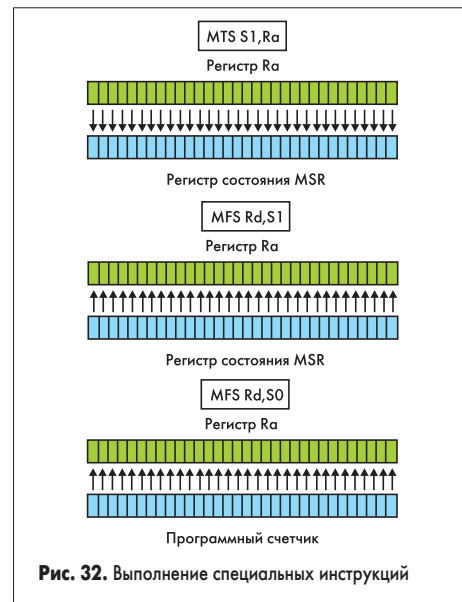


Рис. 32. Выполнение специальных инструкций

Форматы специальных команд представлены в таблице 22.

Инструкция  $MTS S1,Ra$  позволяет загрузить содержимое выбранного регистра общего назначения  $Ra$  в регистр состояния MSR. Данные, записанные в регистр состояния с помощью команды  $MTS$ , вступают в силу через один цикл после исполнения этой инструкции.

Команда  $MFS Rd,Sa$  выполняет копирование содержимого одного из регистров специального назначения  $Sa$  в регистр общего назначения с номером  $d$ . При использовании в инструкции  $MFS$  в качестве второго параметра  $S1$  производится копирование содержимого регистра состояния MSR. Если в качестве второго параметра указано значение  $S0$ , то в регистр общего назначения  $Rd$  записывается содержимое программного счетчика. Рисунок 32 иллюстрирует выполнение специальных инструкций.

Таблица 21. Форматы команд записи данных в выходной канал интерфейса FSL

Поле кода операции	Нулевые разряды	Поле номера регистра	Поле режима вывода	Номер канала интерфейса FSL																Мнемоника	Выполняемая операция														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16			17	18	19	20	21	22	23	24	25	26	27	28	29	30
0 1 1 0 1 1	0 0 0 0 0	a a a a a	1 0 0 0 0	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	PUT Ra,FSLn	Передача информационных данных из регистра Ra в n-й канал интерфейса FSL в режиме блокировки
0 1 1 0 1 1	0 0 0 0 0	a a a a a	1 1 0 0 0	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	nPUT Ra,FSLn	Передача информационных данных из регистра Ra в n-й канал интерфейса FSL без блокировки
0 1 1 0 1 1	0 0 0 0 0	a a a a a	1 0 1 0 0	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	cPUT Ra,FSLn	Передача управляющих данных из регистра Ra в n-й канал интерфейса FSL в режиме блокировки
0 1 1 0 1 1	0 0 0 0 0	a a a a a	1 1 1 0 0	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	ncPUT Ra,FSLn	Передача управляющих данных из регистра Ra в n-й канал интерфейса FSL без блокировки
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31																																Номер разряда микрокоманды			

Таблица 22. Форматы специальных команд

Поле кода операции	Нулевые разряды	Поле номера регистра общего назначения	Поле направления передачи данных и типа регистра специального назначения																												Мнемоника	Выполняемая операция			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28			29	30	31
1 0 0 1 0 1	0 0 0 0 0	a a a a a	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	MTS S1,Ra	Загрузка содержимого регистра общего назначения Ra в регистр состояния MSR
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31																																Номер разряда микрокоманды			
Поле кода операции	Поле номера регистра общего назначения	Нулевые разряды	Поле направления передачи данных и типа регистра специального назначения																												Мнемоника	Выполняемая операция			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28			29	30	31
1 0 0 1 0 1	d d d d d	0 0 0 0 0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	a	MFS Rd,Sa	Загрузка содержимого регистра специального назначения (регистра состояния MSR S1 или программного счетчика S0) в регистр общего назначения Rd	
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31																																Номер разряда микрокоманды			

Таблица 23. Форматы команд записи данных и команд в кэш-память

Поле кода операции	Поле номера первого регистра	Поле номера второго регистра	Поле номера третьего регистра	Поле типа кэш-памяти	Мнемоника	Выполняемая операция
1 0 0 1 0 0	a a a a a	a a a a a	b b b b b	0 0 0 0 1 1 0 1 0 0 0	WIC Ra,Rb	Загрузка содержимого регистра Rb в кэш-память команд.
1 0 0 1 0 0	a a a a a	a a a a a	b b b b b	0 0 0 0 1 1 0 0 1 0 0	WDC Ra,Rb	Загрузка содержимого регистра Rb в кэш-память данных.
0 1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25 26 27 28 29 30 31	Номер разряда микрокоманды	

**Инструкции записи в кэш-память**

Для работы с кэш-памятью в системе команд микропроцессорного ядра MicroBlaze предусмотрены две инструкции, позволяющие загружать содержимое выбранного ре-

гистра общего назначения в кэш-память данных и в кэш-память команд. Эти инструкции используют два параметра. В качестве первого параметра указывается номер регистра общего назначения, который используется для хранения адреса записи. Второй параметр

определяет номер регистра общего назначения, содержимое которого копируется в кэш-память. Форматы команд, предназначенных для работы с кэш-памятью, определены в таблице 23.

Инструкция *WIC Ra,Rb* используется для загрузки содержимого регистра общего назначения *Rb* в кэш-память команд. Инструкция *WDC Ra,Rb* выполняет операцию копирования содержимого регистра *Rb* в кэш-память данных. Рисунок 33 поясняет процесс выполнения этих команд.

На этом завершается описание системы команд микропроцессорного ядра MicroBlaze. Прежде чем приступить к изучению процесса разработки устройств на основе этого модуля, необходимо познакомиться с программными средствами проектирования, составной частью которых является ядро MicroBlaze. Поэтому в следующей публикации цикла будут рассмотрены характеристики и структура пакета Embedded Development Kit (EDK), предлагаемого фирмой Xilinx в качестве основного инструмента разработки и отладки встраиваемых микропроцессорных систем на основе ПЛИС серий FPGA.

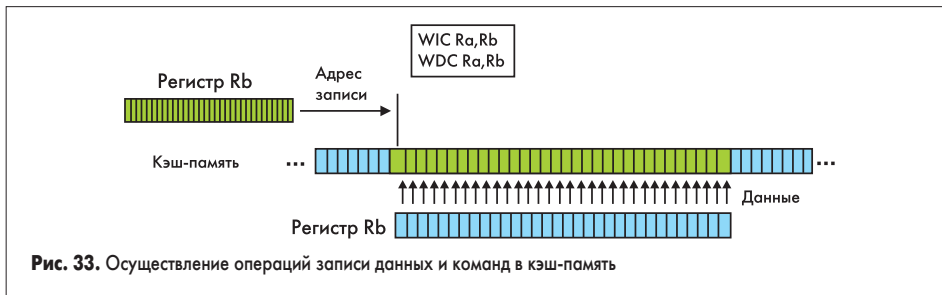


Рис. 33. Осуществление операций записи данных и команд в кэш-память