Обработка-28.

В 2015 году ОАО НИИЭТ начал выполнять ОКР «Обработка-28» в рамках которой по требованиям пользователей предполагается модернизация радиационно-стойкого МК 1874ВЕ7Т, в т.ч. и в вопросах вычислительных мощностей и возможностей ядра. Основные недостатки процессорной части 1874ВЕ7Т:

1. Малый размер адресного пространства. Для обеспечения совместимости с МК, базирующимися на оригинальных MCS-96 архитектурах, ИМС 1874ВЕ7Т имеет такую же как и они адресную область в 64 Кбайт (16-разрядная шина адреса). Минимально требуемый размер на сегодняшний день составляет 16 Мбайт (24-разрядная шина адреса).
2. Отсутствие внутренних областей памяти из которых возможно выполнение программ. Уход в эту область позволил бы обойти ошибки, возникающие во внешней памяти, при обнаружении их кодами коррекции.

Помимо пользовательских предложений, было решено модернизировать ядро следующим образом:

1. Увеличить производительность путем перехода на 32-разрядную шину данных, добавления 32-битного АЛУ и нового набора команд.
2. Ввести подсистему арифметических команд, выполняемых за один такт. Данная подсистема получила название RISC подсистемы, так как однотактовые команды работают с ограниченным количеством регистров.
3. Увеличить разрядность основного для радиационно-стойкого микроконтроллера интерфейса обращений к внешней памяти до 32-бит.
4. Добавить поддержку операций с плавающей точкой IEEE-754 с одинарной (32-бита) и двойной (64-бита) точностью.
5. Расширить адресное пространство до 4 Гбайт (32-разрядная шина адреса).
6. Введение возможности гибкого управления параметрами доступа к внешней памяти с функциями помехоустойчивого кодирования для исправления ошибок доступа.
7. Увеличение количества векторов прерываний и ускорение запуска обработчиков прерываний.
8. Введение механизмов защиты стека от переполнения/опустошения.

Поддержка однотактных команд и увеличенного адресного пространства вводится как для системы команд MCS-96, так и для системы новых 32-битных инструкций. Ранее созданная CISC-платформа позволила сократить время, требуемое для разработки ядра для ОКР «Обработка-28». В настоящее время все вышеописанные функции уже реализованы в Verilog коде, начинается активный процесс его тестирования. Далее будет подробнее описаны особенности нового 32-разрядного ядра.

Подсистема выполнения команд.

Процессорное ядро, используемое в ОКР «Обработка-28» имеет два режима работы ее системы команд:

1. Режим совместимости с MCS-96. В данном режиме микроконтроллер выполняет инструкции системы команд MCS-96. Единственное отличие в том, что часть неиспользуемых в MCS-96 кодов инструкций, в этом режиме используется под новую функциональность. Так, например, в системе команд MCS-96 нет механизма переходов по адресам свыше 64 Кбайт. Для этого в новом ядре во всех режимах работы присутствуют инструкции «очень длинных» переходов и вызовов процедур (VLCALL, VLJMP), которые позволяют осуществить переход на любой адрес адресного пространства. Для поддержки нового размера адресного пространства были внесены изменения и в механизмы выполнения некоторых инструкций MCS-96. Теперь, при вызове прерываний или процедур сохраняется 32-разрядный адрес точки возврата.

2. Режим выполнения 32-разрядных инструкций. В данном режиме вместо части MCS-96 инструкций становятся доступны инструкции работы с 32-разрядными данными. Так, появляются новые инструкции загрузки, арифметические, логические инструкции, инструкции работы со стеком. Также в этом режиме доступны команды работы с байтами системы команд MCS-96.

На рисунках 1 и 2 представлена система команд микроконтроллера в первом и втором режимах работы ядра соответственно.

Код	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	SKIP	CLR	NOT di	NEG di	XCH di	DEC di	EXT	INC di	SHR	SHL	SHRA	XCH in	SHRL	SHLL	SHRAL	NORML
1x	SWC	CLRB	NOTB	NEGB	XCHB di	DECB	EXTB	INCB	SHRB	SHLB	SHRAB	XCHB in	***	***	***	***
2x	SJMP								SCALL							
3x	JBC								JBS							
	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
4x	AND 3op				ADD 3op				SUB 3op				MULU 3op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
5x	ANDB 3op				ADDB 3op				SUBB 3op				MULUB 3op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
6x	AND 2op				ADD 2op				SUB 2op				MULU 2op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
7x	ANDB 2op				ADDB 2op				SUBB 2op				MULUB 2op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
8x	OR				XOR				CMP				DIVU *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
9x	ORB				XORB				CMPB				DIVUB *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
Ax	LD				ADDC				SUBC				LDBZE			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
Bx	LDB				ADDCB				SUBCB				LDBSE			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
Cx	ST	BMOV	ST		STB	CMPL	STB		PUSH				POP	BMOVI	POP	
	di	-	in	ix	di	di	in	ix	di	im	in	ix	di	-	in	ix
Dx	JNST	JNH	JGT	JNC	JNVT	JNV	JGE	JNE	JST	JH	JLE	JC	JVT	JV	JLT	JE
Ex	DJNZ	DJNZW	TIJMP	BR in	VLCALL	F-инстр.	VLJMP	LJMP	***	***	***	***	DPTS	EPTS	прерыв.	LCALL
Fx	RET	TRAP1	PUSHF	POPF	PUSHA	POPA	MOD	TRAP	CLRC	SETC	DI	EI	CLRVT	NOP	*	RST

Обозначение	Прямая		Непосредственная		Косвенная (1)		Индексная S/L (1)	
DIV	4	FE 8C	5	FE 8D	4	FE 8E	5/6	FE 8F
DIVB	4	FE 9C	4	FE 9D	4	FE 9E	5/6	FE 9F
MUL (2 ops)	4	FE 6C	5	FE 6D	4	FE 6E	5/6	FE 6F
MUL (3 ops)	5	FE 4C	6	FE 4D	5	FE 4E	6/7	FE 4F
MULB (2 ops)	4	FE 7C	4	FE 7D	4	FE 7E	5/6	FE 7F
MULB (3 ops)	5	FE 5C	5	FE 5D	5	FE 5E	6/7	FE 5F

Рисунок 1 – Система команд в режиме MCS-96

Код	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	SKIP	ECLR	ENOT	ENEG	ELDS	EDEC	EXT	EINC	SHR	SHL	SHRA	XCH in	ESHRL	ESHLL	ESHRAL	NORML
			di	di	di	di		di	SHRB	SHLB	SHRAB	XCHB	di	di	di	
1x	SWC	CLRB	NOTB	NEGB	XCHB di	DECB	EXTB	INCB				in	EST shot ix pc			
2x	SJMP								SCALL							
3x	JBC								JBS							
	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
4x	AND 3op				ADD 3op				SUB 3op				MULU 3op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
5x	ANDB 3op				ADDB 3op				SUBB 3op				MULUB 3op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
6x	EAND 2op				EADD 2op				ESUB 2op				EMULU 2op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
7x	ANDB 2op				ADDB 2op				SUBB 2op				MULUB 2op *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
8x	EOR				EXOR				CMP				DIVU *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
9x	ORB				XORB				CMPB				DIVUB *			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	di	im	in	ix
Ax	ELD				EADDC				ESUBC				ELD long			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	ix pc	ix pc	ix pc	ix pc
Bx	LDB				ADDCB				SUBCB				ELD shot			
	di	im	in	ix	di	im	in	ix	di	im	in	ix	ix pc	ix pc	ix pc	ix pc
Cx	ST	BMOV	EST		STB	ECMPL	STB		EPUSH				EPOP	BMOVI	EPOP	
	di	-	in	ix	di	di	in	ix	di	im	in	ix	di	-	in	ix
Dx	JNST	JNH	JGT	JNC	JNVT	JNV	JGE	JNE	JST	JH	JLE	JC	JVT	JV	JLT	JE
Ex	DJNZ	DJNZW	TIJMP	BR in	VLCALL	F-инстр.	VLJMP	LJMP	EST long ix pc				DPTS	EPTS	прерыв.	LCALL
Fx	RET	TRAP1	PUSHF	POPF	PUSHA	POPA	MOD	TRAP	CLRC	SETC	DI	EI	CLRVT	NOP	*	RST

Обозначение	Прямая		Непосредственная		Косвенная (1)		Индексная S/L (1)	
DIV	4	FE 8C	5	FE 8D	4	FE 8E	5/6	FE 8F
DIVB	4	FE 9C	4	FE 9D	4	FE 9E	5/6	FE 9F
MUL (2 ops)	4	FE 6C	5	FE 6D	4	FE 6E	5/6	FE 6F
MUL (3 ops)	5	FE 4C	6	FE 4D	5	FE 4E	6/7	FE 4F
MULB (2 ops)	4	FE 7C	4	FE 7D	4	FE 7E	5/6	FE 7F
MULB (3 ops)	5	FE 5C	5	FE 5D	5	FE 5E	6/7	FE 5F

Рисунок 2 – Система команд в режиме выполнения инструкций работы с 32-битными данными. Оранжевым цветом выделены новые инструкции.

Переход из одного режима работы ядра в другой осуществляется инструкцией SWC. В систему команд была введена еще одна специальная инструкция – команда конца инициализации EINIT. После первого запуска этой команды становятся недоступными для модификации часть системных регистров микроконтроллера – регистры настроек доступа к внешней шине, регистры определения границ стека, регистры определения начальных адресов подпрограмм обработчиков прерываний.

Максимальная длина команд во всех режима работы ядра осталась раной 7 байт, но благодаря масштабируемости Verilog описания возможно введение инструкций и большей длины.

Увеличение разрядности шины данных и появление 32-разрядного АЛУ – это только одно из наиболее очевидных решений на пути увеличения производительности. Вторым направлением было введение команд, которые могли бы выполняться за один такт. Так как сделать это для всего регистрового файла, к которому в архитектуре MCS-96 относится ОЗУ, технически невозможно, была определена область адресного пространства, для которого однотактные команды будут доступны. Таким образом первые адреса адресного пространства, начиная с адреса 0000_0004h, и кончая адресом 0000_0044h, образуют область 32-разрядных регистров R0-R15, при попадании операндов в которую используется свой набор внутренних микроопераций, что позволяет существенно ускорить выполнение команд. Так как определить попадают ли операнды в область R0-R15 нужно в тот же такт, что и загружается команда в декодер микроопераций, то такое ускорение возможно сделать только для команд использующих прямую адресацию, так как только при таком виде адресации конечные адреса операндов определены в самой команде. Если при

прямой адресации операнды и результат попадают в область R0-R15, то выполнение такой команды происходит с помощью специальной подсистемы (назовем ее RICS, так как сокращен набор адресов, с которой она работает) за один такт. На рисунках 1 и 2 команды, которые имеют свой т.н. RISC эквивалент выполнения, выделены синим цветом. Как видно из рисунков, RISC подсистема работает во всех режимах работы ядра. Таким образом, например, команда ADD 20h,22h из системы команд MCS-96 будет выполняться за один такт, а команда ADD 60h,22h – за четыре. Особый случай для команд с тремя операндами и командами сдвига. Для первых существует два типа ускорения: в первом случае, если все три операнда попадают в область регистров, то выполнение происходит за один такт, а в случае, если в область регистров попадают только входные операнды, а результат необходимо поместить в область ОЗУ, то выполнение команды происходит за два такта. Для команд сдвига помимо условия нахождения операнда в области R0-R15 существует второе условие ускорения – количество сдвигов должно быть указано в самой команде.

Механизм RISC подсистемы дает дополнительные возможности по оптимизации кода по объему/скорости выполнения, так как для каких-то случаев позволяет минимизировать количество команд (для выполнение арифметических действий не требуется пересылка из области ОЗУ в область регистров R0-R15 дополнительными командами), а для других – минимизировать время выполнения. Архитектуру микроконтроллера можно назвать CISC+RISC архитектурой, при этом достоинством ее является то, что CISC и RISC команды имеют одинаковый код.

Еще одним дополнительным блоком ядра микроконтроллера, позволяющим увеличить производительность, но уже в операциях с плавающей запятой, является блок FPU, реализующий стандарт IEEE-754. Для выполнения новых команд, были введены, так называемые F-инструкции. Блок FPU работает со своим набором 32-разрядных регистров FR0-FR1, адресация к которым доступна как специальными инструкциями FLD, FST по коротким адресам, так и по адресам общего адресного пространства. Блок FPU поддерживает операции с одинарной (32-бита) и с двойной (64-бита) точностью. Вычисление результата блоком FPU возможно каждый такт.

Подсистема выборки команд.

Для обеспечения максимальной производительности необходимо иметь такую подсистему выборки команд, чтобы она как можно быстрее подготавливала инструкцию к исполнению. В условиях, когда длина команд нефиксированная, увеличить производительность только увеличением разрядности интерфейсов доступа к командам невозможно. Например, выбрав 64 бит из области команд (при длине команд от 8 до 56 бит) за один такт определить все находящиеся в этом блоке команды технически невозможно (конечно, разумными для этой задачи ресурсами). Но, с другой стороны, для некоторого набора команд при некоторых условиях (именно условиях, а не требованиях), увеличение разрядности интерфейса позволяет ускорить выполнение программ. Исходя из этого подсистема выборки команд в ИМС, создаваемой в ходе ОКР «Обработка-28», имеет ряд улучшений, относительно других контроллеров архитектуры MCS-96.

1. Появление так называемой истории команд на 32 (в текущей версии) инструкции. Если осуществляется переход в рамках последних выполняемых 32 инструкций, то блок выборки команд не обращается заново к внешней памяти или внутренней PSRAM, а загружает очередь команд из сохраненной ранее области. Данный механизм отключаемый. Размер области истории команд был выбран таким первоначально и является обсуждаемым с потенциальными потребителями.
2. Увеличена очередь команд. Размер ее составляет 16 инструкций.
3. Введен новый 32-разрядный режим доступа к внешней памяти команд. Его особенности будут описаны ниже.

4. Введен механизм обращения к различным областям внешней памяти с индивидуальными настройками параметров доступа. Микроконтроллер имеет набор из восьми пар 32-разрядных регистров выбора границ окон (ADDRSELx_LO, ADDRSELx_HI) и восьми управляющих параметрами доступа регистров (BUSCONx). Для каждой из областей возможны индивидуальные настройки следующих параметров: количество тактов ожидания при обращении к шине, активный уровень сигнала CS (chip select), разрешение сигнала CS, использование помехоустойчивого кодирования, тип доступа (8, 16, 32-разрядный).

По сравнению с архитектурой MCS-96 микроконтроллер «Обработка-28» имеет еще один режим доступа к внешней шине – 32-разрядный. В данном режиме подсистема выборки команд получает сразу 4 байта информации и начинается процесс их декодирования. Длительность такого процесса составляет от одного до 4 тактов. В случае, если команды выравнены по адресам кратным 4 и в блоке информации присутствует только одна команда (4 байтная), то декодирование и размещение такой инструкции в очереди команд будет самым быстрым. Самое долгое декодирование будет в случае, когда из внешней памяти выбираются четыре однобайтных инструкции. В этом случае, их размещение в очереди команд займет четыре такта. Механизм работы подсистемы выборки аналогичен процессу распаковки информации: чем больше в полученном блоке информации (в данном случае инструкций), тем дольше он идет. Этот процесс автоматический, и наглядно представлен на рисунке 3 –длительность внешнего сигнала RD пропорционально времени размещения команд в очереди. Возможность ускорения выборки из памяти дает еще одно направление в процессе оптимизации пользовательского кода.

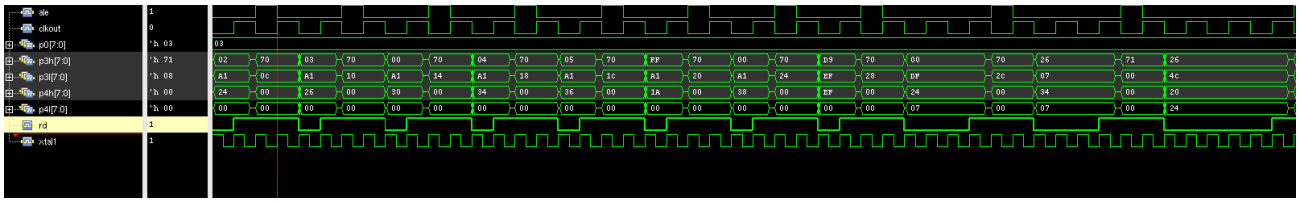


Рисунок 3 – Обращение к внешней памяти по 32-разрядному интерфейсу

В отличие от оригинальных микроконтроллеров архитектуры MCS-96, ИМС «Обработка-28» имеет область памяти типа ОЗУ (PSRAM), из которой возможно выполнение команд. В данной области возможно хранение обработчиков прерываний, вызванных ошибками в командах или данных, размещенных во внешней памяти в случае использования помехоустойчивого кодирования. Доступ к данному виду памяти осуществляется по 32-разрядной шине и его скорость также зависит от набора команд, как и для режима внешней памяти.

Адресное пространство.

Микроконтроллер «Обработка-28» имеет 4 Гбайт адресного пространства. Карта памяти представлена на рисунке 4.



Рисунок 4 – Адресное пространство микроконтроллера

Старт микроконтроллера происходит с адреса 0000_7000h.

Система команд MCS-96 не предусматривает команд переходов и способов адресации для области памяти свыше 64 Кбайт. Для поддержки функциональности новой адресной области были проведены следующие мероприятия:

1. Для осуществления переходов и запусков подпрограмм в обоих режимах работы ядра появились команды «очень длинных» переходов VLCALL, VLJMP. Также при вызовах подпрограмм инструкциями SCALL, LCALL, при переходах на подпрограммы обработчиков прерываний в стеке сохраняется полный 32-разрядный адрес.
2. Для доступа к данным вводится механизм адресации с помощью указателей страниц данных (data page pointers – DPP).

Микроконтроллер имеет 4 указателя страниц данных. Они позволяют расширить адрес в любом из режимов работы ядра до 32-разряда. В случае работы ядра в режиме MCS-96 работают только два указателя из 4-х. Формирование адреса в этом случае представлено на рисунке 5.

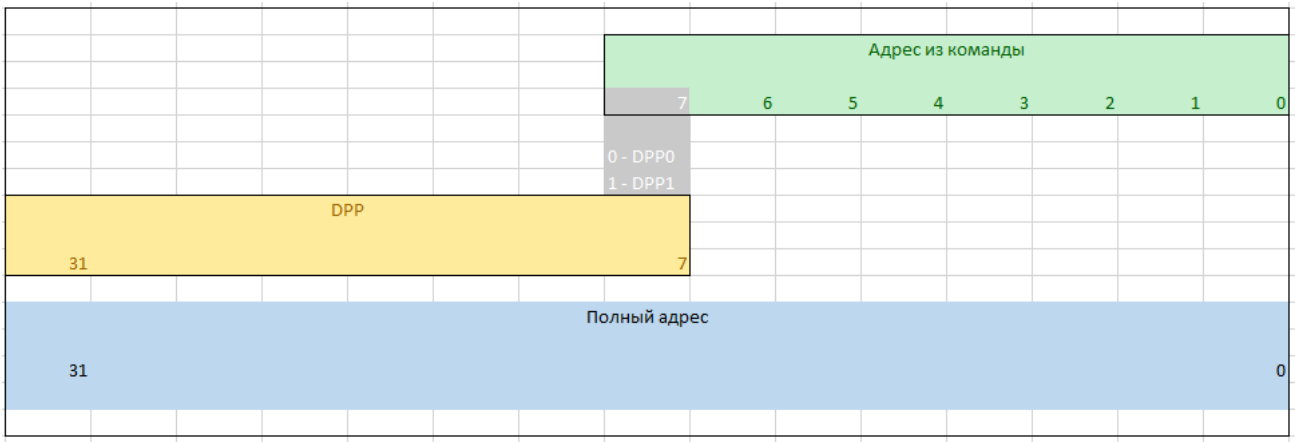


Рисунок 5 – Формирование адреса операнда в случае работы ядра в режиме MCS-96.

Особенностью использования указателей является то, что они расширяют адрес только при микрооперациях прямой адресации. Какой из указателей DPP будет применяться при формировании адреса, выбирается старшим битом адреса из команды. Рассмотрим это на примере. В случае выполнения команды ADD 60h, 62h запускаемые микрооперации будут выглядеть следующим образом:

1. Происходит чтение по прямому адресу значения ячейки 62h. Значение попадает в АЛУ.
2. Происходит чтение по прямому адресу значения ячейки 60h. Значение попадает в АЛУ.
3. Результат операции после вычисления по прямому адресу заносится в ячейку 60h.

В данном случае во всех трех стадиях выполнения команды используется прямая адресация, соответственно во всех них будет использоваться DPP. В случае выполнения команды ADD 60h, [62h] запускаемые микрооперации будут выглядеть следующим образом:

1. Происходит чтение по прямому адресу значения ячейки 62h.
2. Происходит чтение с использованием косвенной адресации по адресу, взятому из ячейки 62h. Это значение попадает в АЛУ.
3. Происходит чтение по прямому адресу значения ячейки 60h. Значение попадает в АЛУ.
4. Результат операции после вычисления по прямому адресу заносится в ячейку 60h.

В данном случае DPP используется при адресации всегда, кроме шага 2. Во всех случаях, когда адрес берется из ячейки памяти, указатели страниц не используются. Особенностью адресации в данном режиме работы ядра является то, что косвенной адресации доступно только 64Кбайт, как и в MCS-96. То есть ячейка с адресом может находится в любом месте адресного пространства, но считается оттуда только 16 бит конечного адреса.

Для режима работы с 32-разрядными данными применяются все четыре указателя страниц. Формирование адреса при этом представлено на рисунке 6.

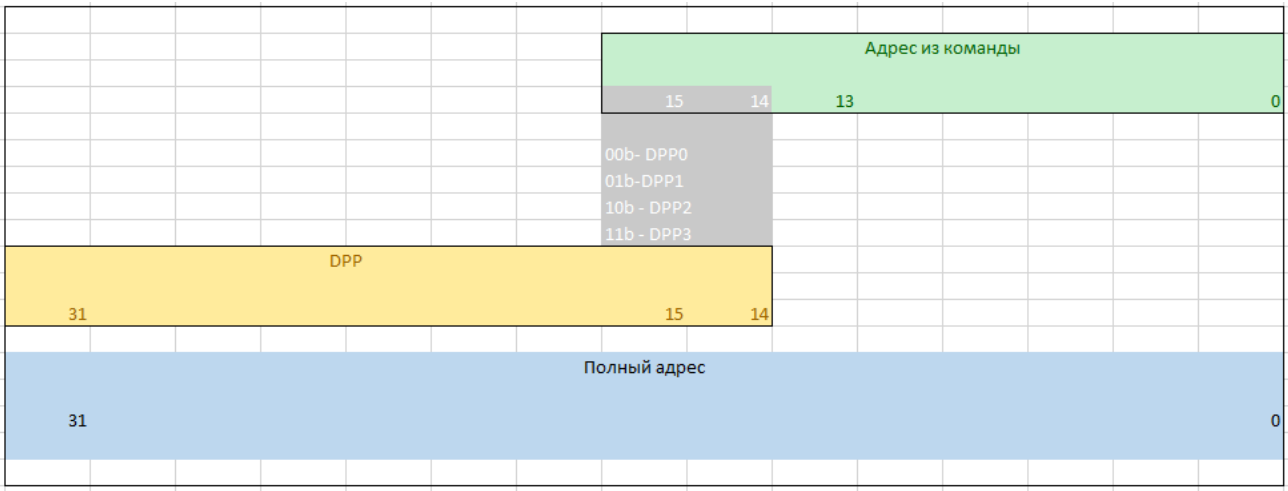


Рисунок 6 – Формирование адреса в режиме работы с 32-разрядными данными.

Особенности работы в данном режиме: определяют, какой будет использоваться указатель, уже два бита адреса команды; при операции чтения или записи по адресу, заданному косвенно, из ячейки памяти считывается уже 32-бит адреса. То есть и ячейка с указателем, и сама ячейка, откуда берутся данные для операции – все это может находится в любом месте адресного пространства.

Указатели на страницу данных не используются для адресов, участвующих в прямой адресации, меньших 0000_0060h. В этой области находятся регистры R0-R15 и сами указатели DPP.

Работа с прерываниями.

Нововведения в работе с прерываниями в ИМС «Обработка-28» связаны со следующими особенностями схемы:

1. Увеличение числа периферийных блоков и рост их функциональности привели к необходимости увеличить число векторов прерывания. В данной редакции ядра их 64 штуки. Управляются прерывания посредством двух 32-х разрядных регистров INT_MASK и INT_MASK2. Запросы на прерывания ожидают обработки в регистрах INT_PEND и INT_PEND2.
2. Появление в архитектуре новых служебных прерываний: прерывания по ошибкам внешнего интерфейса, прерывания по переполнению и опустошению стека, прерывания по ошибкам доступа к регистрам FPU.
3. Изменение области расположения векторов прерывания. Связано с появлением помехоустойчивого кодирования интерфейса внешней памяти. В архитектуре MCS-96 вектора располагаются в той же области, что и команды. Если используется режим работы с внешней памятью, то и вектора будут находится в ней. Это могло бы привести к парадоксальной ситуации: для того, чтобы правильно выйти на обработчик прерывания по ошибке внешнего интерфейса, микроконтроллер должен считать адрес перехода из той же внешней памяти. В ИМС «Обработка-28» адреса обработчиков прерываний хранятся в регистрах (INTx_ADDR), находящихся в области SFR. По умолчанию, адреса настроены на расположение обработчиков во внешней памяти, но могут быть и переназначены на область PSRAM (которую тоже предполагается защитить от сбоев). Регистры INTx_ADDR должны быть установлены в инициализационной части программы пользователя и не могут быть изменены после

выполнения команды EINIT. Подход с расположением векторов в области SFR также позволил ускорить переход на обработчик прерывания, так как ликвидировалась одно (с точки зрения пользовательской программы не несущее полезной функциональности) обращение к внешней памяти: после обработки запроса микроконтроллер сразу выходит на нужный адрес, в котором содержатся команды.

В микроконтроллере «Обработка-28» произошли изменения и в механизме обработки прерываний блоком периферийных транзакций (PTS). Из всего многообразия режимов работы PTS архитектуры MCS-96, в новом микроконтроллере остались только режимы одиночной и блочной передачи. Данные режимы являются универсальными и позволяют вести пересылку данных между любыми ячейками всего адресного пространства. Еще одним нововведением в работе блока PTS является появление возможности выбора для каждого сеанса обмена типа приоритета: приоритет ядра или приоритет PTS. В первом случае, блок периферийных транзакций будет отслеживать моменты, когда не используется внутренняя память (например, запущена микрооперация вычисления в АЛУ или простой шины вследствие неготовности команды для загрузки в декодер микроопераций) и в этим моменты вести свой обмен, во втором – PTS остановит выполнение команд ядром и выполнит обмен данными максимально быстро.