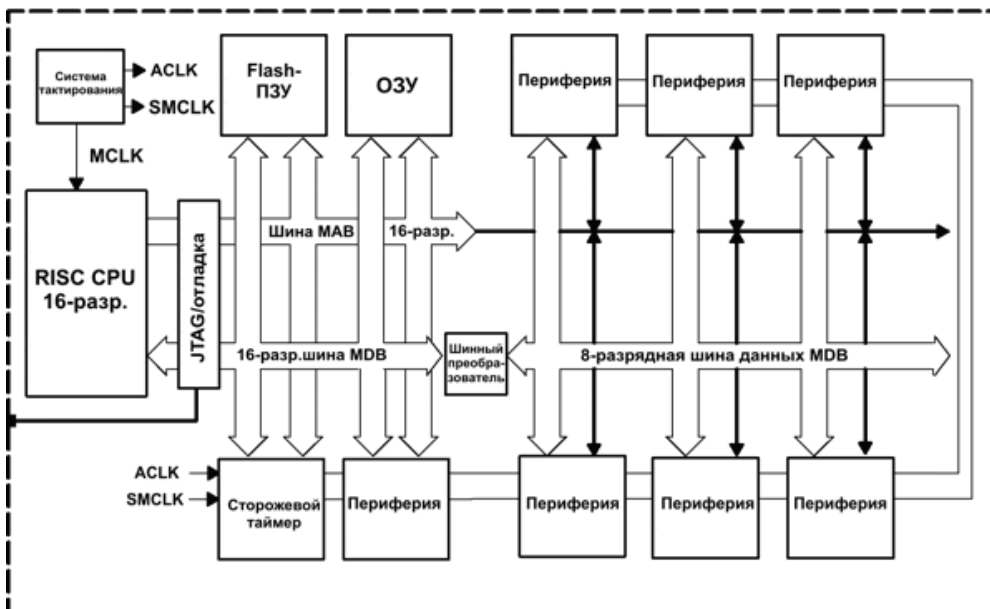


## Ответы на теоретические вопросы

### 1. Обзор а1. Обзор архитектуры микроконтроллеров MSP430x1xx. Система тактирования.

Система тактирования разработана специально для использования в устройствах с питанием от батарей. Вспомогательная низкочастотная система тактирования (ACLK) работает непосредственно от обычного 32 кГц часового кристалла. Модуль ACLK может использоваться в качестве фоновой системы реального времени с функцией самостоятельного «пробуждения». Интегрированный высокоскоростной осциллятор с цифровым управлением (DCO) может быть источником основного тактирования (MCLK) для ЦПУ и высокоскоростных периферийных устройств. Модуль DCO становится активным и стабильным менее чем через 6 мкс после запуска. Решения на основе архитектуры MSP430 позволяют эффективно использовать высокопроизводительное 16-разрядное RISC CPU в очень малые промежутки времени:

- низкочастотная вспомогательная система тактирования обеспечивает работу микроконтроллера в режиме ультранизкого потребления мощности;
- активизация основного высокоскоростного модуля тактирования позволяет выполнить быструю обработку сигналов.



### 2. Обзор архитектуры микроконтроллеров MSP430x1xx. Адресное пространство.

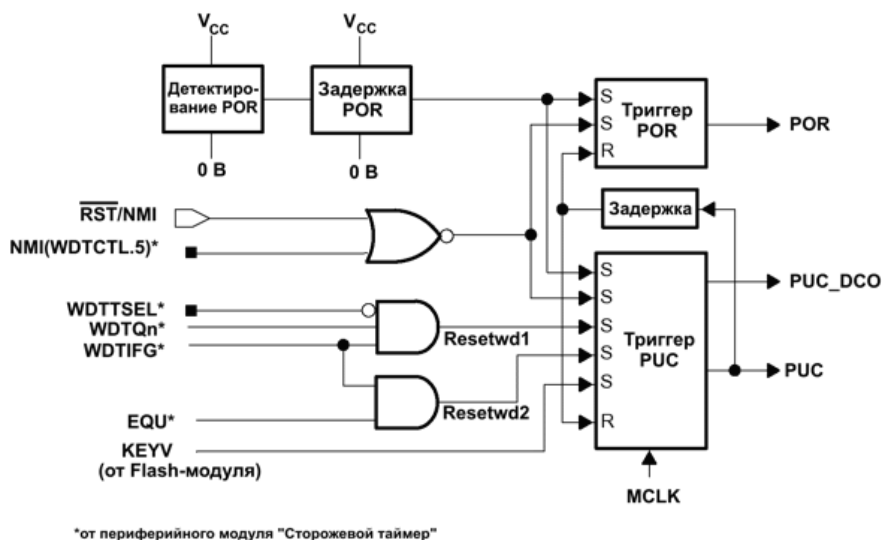
Семейство MSP430 имеет фон-Неймановскую архитектуру с единым адресным пространством для регистров специального назначения (SFR), периферии, ОЗУ и Flash-памяти программ, в соответствии с рис.1.2. Конкретное распределение памяти можно узнать из справочных данных на интересующее устройство. Доступ к программному коду выполняется всегда по четным адресам. Данные могут быть доступны как байты или как слова.

Общий объем адресуемой памяти составляет 64 кБ, с учетом предполагаемого расширения.

Доступ		
0FFFFh	Таблица векторов прерываний	Слово/байт
0FFE0h		
0FFDFh		
	Flash-ПЗУ	Слово/байт
	ОЗУ	Слово/байт
0200h		
01FFh	16-разрядные периферийные модули	Слово
0100h		
0FFh	8-разрядные периферийные модули	Байт
010h		
0Fh	Регистры специального назначения	Байт
0h		

### 3. Системный сброс и инициализация MSP430x1xx.

В схеме системного сброса, показанной на рис. 2.1 источниками сброса могут быть сигналы сброса при включении (POR) и очистки при включении (PUC). Различные события и исходные условия определяют, какой именно из этих сигналов будет сгенерирован.



Сигнал POR сбрасывает устройство. Он может быть сгенерирован в следующих двух случаях:

- включение устройства;
- появление сигнала низкого уровня на выводе RST/NMI, когда он сконфигурирован как вход сигнала «сброса»;
- низкий уровень напряжения питания.

Сигнал PUC генерируется всегда при появлении сигнала POR, сигнал POR не генерируется сигналом PUC. Следующие события приводят к появлению сигнала PUC:

- сигнал POR;
- срабатывание «сторожевого» таймера (только если сторожевой таймер активирован);
- произошло нарушение ключа безопасности «сторожевого» таймера;
- произошло нарушение ключа безопасности Flash-памяти;
- попытка выборки команды из памяти в диапазоне адресов 0h...01FFh.

После системного сброса пользовательское программное обеспечение должно **инициализировать** MSP430 в соответствии с требованиями конкретного приложения. Необходимо выполнить следующие действия:

- Инициализировать указатель стека SP (как правило, указывается вершина ОЗУ)
- Инициализировать сторожевой таймер в зависимости от требований приложения
- Сконфигурировать периферийные модули в зависимости от требований приложения

Дополнительно можно оценить состояние флагов сторожевого таймера, флэш-памяти и неисправности осциллятора для определения источника сброса.

#### 4. Прерывания MSP430x1xx. Маскируемые и немаскируемые прерывания.

Приоритеты прерываний показаны на рис. 2.4. Приоритеты определяются порядком расположения модулей в соединяющей их цепи. Чем ближе модуль к ЦПУ/NMIRS, тем выше его приоритет.

Прерывания делятся на три типа:

- Системное (системный сброс)
- Немаскируемое (NMI)
- Маскируемое

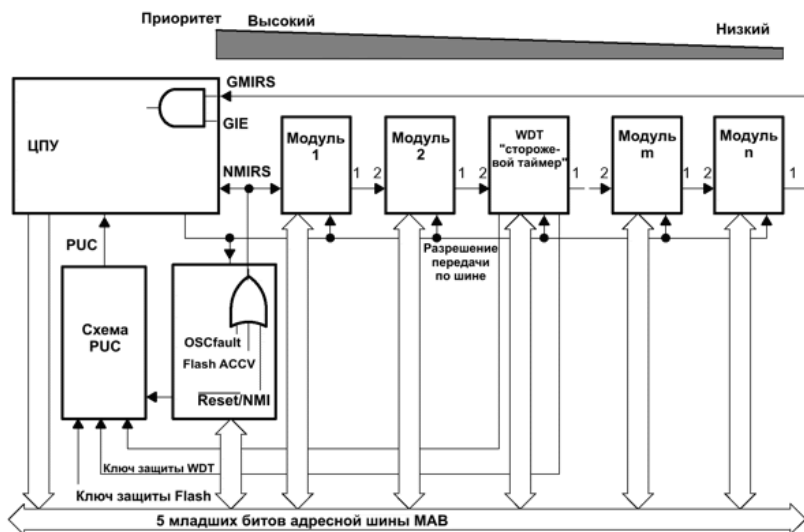


Рис.2-4 Приоритеты прерываний

**Немаскируемые прерывания** NMI не маскируются общим битом разрешения прерываний (GIE), но могут управляться индивидуальными битами включения прерывания (ACCVIE, NMIIIE, OFIE). Когда происходит немаскируемое прерывание NMI, все биты разрешения NMI-прерываний автоматически сбрасываются. Выполнение программы продолжается с адреса, содержащегося в векторе немаскируемого прерывания (0FFFCh). Программное обеспечение пользователя должно установить необходимые биты NMI-прерывания, чтобы оно было разрешено вновь. Блок-схема источников NMI-прерываний показана на рис.2.5.

Немаскируемое прерывание NMI может быть вызвано тремя событиями:

- Появление фронта сигнала на выводе RST/NMI
- Появление неисправности тактового генератора

- Нарушение доступа к флэш-памяти

**Маскируемые прерывания** вызываются периферийными устройствами, имеющими возможность вызова прерываний, включая ситуацию переполнения сторожевого таймера в активном режиме. С помощью индивидуальных битов разрешения прерывания можно отключать источники прерываний как по отдельности, так и все сразу с использованием общего бита разрешения всех прерываний (GIE) в регистре статуса (SR).

## 5. Обработка прерываний MSP430x1xx. Векторы прерываний.

Если периферийное устройство запросило прерывание и включены биты общего разрешения прерываний GIE и индивидуальный бит разрешения прерывания от этого устройства, будет вызвана процедура обработки прерывания. Для вызова немаскируемого (NMI) прерывания достаточно установки только индивидуального бита разрешения прерывания.

**Векторы прерываний** и стартовые адреса расположены в адресном диапазоне с 0FFFFh по 0FFE0h, как показано в таблице 2.1. Вектор программируется пользователем с помощью указания 16-разрядного стартового адреса соответствующей процедуры обработки прерывания.

Время задержки вызова прерывания составляет 6 машинных циклов. Обработка запроса на прерывание производится в следующей последовательности: 1. Ожидается завершение команды, исполняемой в данный момент. 2. Содержимое счетчика команд PC, указывающего на следующую команду, помещается в стек. 3. Содержимое регистра состояния SR помещается в стек. 4. Если за время выполнения последней команды было сформировано несколько запросов на прерывание, то выбирается прерывание с наибольшим приоритетом. 5. Если прерывание имеет один источник, то флаг прерывания автоматически сбрасывается. Если прерывание может генерироваться несколькими источниками, то флаги прерывания остаются установленными для последующей обработки в программе. 6. Регистр состояния SR очищается. В результате процессор переходит из режима пониженного потребления в активный режим. Поскольку бит GIE сбрасывается, последующие прерывания запрещаются. 7. Содержимое вектора прерывания загружается в счетчик команд PC и начинается выполнение процедуры обработки прерывания, расположенной по этому адресу.

## 6. Режимы работы MSP430x1xx.

Семейство MSP430 разработано для приложений с ультранизким потреблением мощности и имеет различные режимы работы, показанные на рис.2.10.

Режимы работы учитывают три различные потребности:

- ультранизкое потребление
- скорость и пропускную способность
- минимизацию потребления тока конкретной периферией

Типичное потребление тока микроконтроллерами семейства MSP430 показано на рис.2.9.

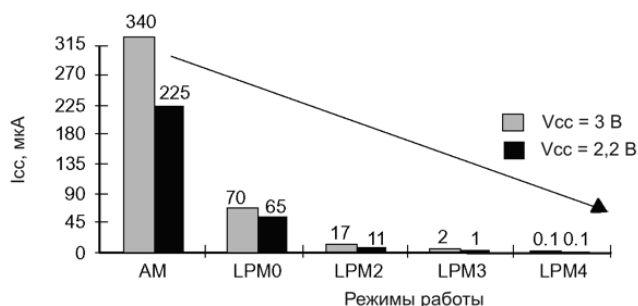


Рис.2-9 Типичное потребление тока устройствами 13х и 14х в зависимости от режима работы

SCG1	SCG0	OSCOFF	CPUOFF	Режим	Состояние ЦПУ и тактовых сигналов
0	0	0	0	Активный	ЦПУ активно, все разрешённые тактовые сигналы активны
0	0	0	1	LPM0	ЦПУ, MCLK выключены SMCLK, ACLK активны
0	1	0	1	LPM1	ЦПУ, MCLK выключены. Генератор постоянного тока выключен, если DCO не используется для формирования SMCLK. ACLK активен
1	0	0	1	LPM2	CPU, MCLK, SMCLK, DCO выключены. Генератор постоянного тока не выключается. ACLK активен
1	1	0	1	LPM3	CPU, MCLK, SMCLK, DCO выключены. Генератор постоянного тока выключен. ACLK активен
1	1	1	1	LPM4	ЦПУ и все тактовые сигналы выключены

**Рис. 2.8.** Режимы работы основной системы тактирования MSP430x2xx.

Режимы низкого энергопотребления 0-4 конфигурируются с помощью битов CPUOFF, OSCOFF, SCG0 и SCG1 в регистре статуса. Преимущество включения битов управления режимом CPUOFF, OSCOFF, SCG0 и SCG1 в состав регистра статуса SR состоит в том, что текущий режим работы может быть сохранен, путем помещения содержимого SR в стек во время работы процедуры обработки прерывания. Выполняемая программа возвращается к предыдущему режиму работы, если сохраненное содержимое регистра SR не было изменено процедурой обработки прерывания. Выполнение программы может продолжиться в другом рабочем режиме, если процедура обработки прерывания изменит значение регистра SR в стеке. Обращение к битам управления режимом и стеку может производиться с помощью любой команды.

При изменении любого бита управления режимом, выбранный режим работы активизируется немедленно. При отключении любой системы тактирования, блокируются также периферийные устройства, работающие от этой системы. Периферийные устройства также могут отключаться с помощью соответствующих им индивидуальных управляющих регистров. Состояние всех выводов портов ввода/вывода и ячеек ОЗУ остается неизменным. «Пробуждение» возможно через все разрешенные прерывания.

## 7. 16-разрядное RISC CPU MSP430x1xx.

ЦПУ включает возможности, специально созданные для современных технологий программирования, таких как вычисляемое ветвление, обработка таблиц и использование языков высокого уровня, подобных языку С. ЦПУ может выполнять адресацию в полном адресном диапазоне без использования страниц памяти.

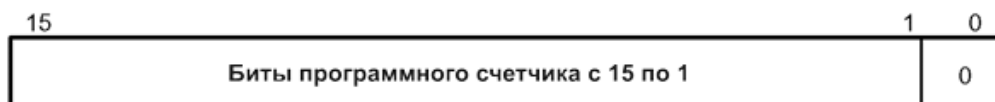
**ЦПУ обладает следующими возможностями:**

- RISC-архитектура с 27 командами и 7 режимами адресации;
- Ортогональная архитектура, при которой каждая команда пригодна для каждого режима адресации;
- Полный доступ ко всем регистрам, включая программный счетчик, регистры статуса и указатель стека;
- Однотактные регистровые операции;
- Большой 16-разрядный регистровый файл, уменьшающий количество обращений к памяти;
- 16-разрядная адресная шина, обеспечивающая прямой доступ и ветвление во всем диапазоне памяти;
- 16-разрядная шина данных, позволяющая напрямую манипулировать параметрами шириной в слово;
- Генератор констант немедленно предоставляет шесть используемых наиболее часто значений, уменьшая размер кода;
- Прямой обмен между ячейками памяти без промежуточной записи в регистр;
- Команды и адресация в форматах «слово» и «байт».

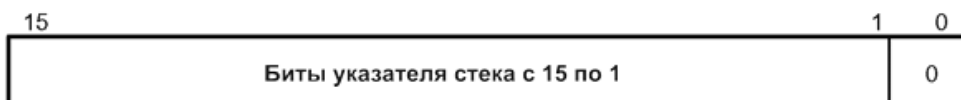
## 8. Регистры ЦПУ.

ЦПУ включает шестнадцать 16-разрядных регистров. Регистры R0, R1, R2 и R3 имеют специальное назначение. Регистры с R4 по R15 являются рабочими регистрами общего назначения.

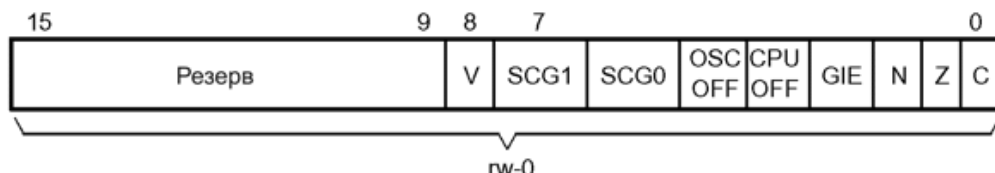
**16-разрядный программный счетчик (PC/R0)** указывает на следующую команду, которая будет выполняться. Каждая команда состоит из четного числа байтов (два, четыре или шесть), поэтому PC инкрементируется соответственно. Команды доступа в адресном пространстве 64 кБайт выполняются к границам слов, поэтому PC выравнивается к четным адресам. На рис.3.2 показана организация программного счетчика.



**Указатель стека (SP/R1)** используется ЦПУ для хранения адресов возврата из подпрограмм и прерываний. Стек основан на предекрементной постинкрементной схеме. Кроме того, указатель стека SP может использоваться со всеми командами и во всех адресных режимах. На рис.3.3 показана организация SP. Указатель стека SP инициализируется в ОЗУ пользователем и выравнивается к четным адресам.



**Регистр статуса (SR/R2)**, используемый как регистр источника или получателя, может адресоваться в регистровом режиме только с помощью команд-слов. Прочие комбинации режимов адресации используются для поддержки генератора констант. На рис.3.6 показаны биты регистра статуса SR.



Шесть обычно используемых констант генерируются с помощью **регистров R2 и R3 генератора констант**, что исключает необходимость использования дополнительного 16-разрядного слова в программном коде. Константы выбираются путем изменения режима адресации (As) регистра-источника.

**Двенадцать регистров с R4 по R15** являются регистрами общего назначения. Все эти регистры могут быть использованы в качестве регистров данных, указателей адресов или индексных значений и доступны с помощью команд работы с байтами или словами, как показано на рис.3.7.



## 9. Режимы адресации.

Семь режимов адресации для операнда источника и четыре режима адресации для операнда назначения могут адресовать полное адресное пространство без исключений. В таблице 3.3 приводится конфигурация битов для режимов As (источник) и Ad (назначение).

As/Ad	Режим адресации	Синтаксис	Описание
00 / 0	Регистровый режим	Rn	Содержимое регистра является операндом
01 / 1	Индексный режим	X(Rn)	Значение (Rn+X) указывает на операнд. X сохранен в следующем слове
01 / 1	Символьный режим	ADDR	Значение (PC+X) указывает на операнд. X сохранен в следующем слове. Использован индексный режим X(PC)
01 / 1	Абсолютный (безусловный) режим	&ADDR	Слово, следующее за командой, содержит абсолютный адрес. X сохранен в следующем слове. Использован индексный режим X(SR)
10 / -	Косвенный регистровый режим	@Rn	Содержимое Rn использовано как указатель на операнд
11 / -	Косвенный автоинкремент	@Rn+	Содержимое Rn использовано как указатель на операнд. Содержимое Rn впоследствии увеличивается на 1 для байтовых команд и на 2 для команд-слов.
11 / -	Прямой (непосредственный) режим	#N	Слово, следующее за командой, содержит непосредственную константу N. Использован косвенный автоинкрементный режим @PC+

## 10. Система команд. Команды с двойным операндом.

В общей сложности набор команд ЦПУ MSP430 включает в себя 27 команд ядра и 24 эмулируемых команды.

На рис.9 показана структура формата команды с двойным операндом.

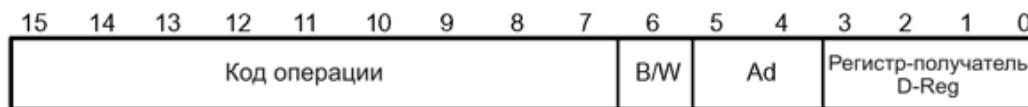


В таблице 3.11 приведено описание и перечень команд с двойным операндом.

Мнемоника	S-Reg, D-Reg	Операция	Биты статуса			
			V	N	Z	C
MOV(.B)	src,dst	src -> dst	-	-	-	-
ADD(.B)	src,dst	src + dst -> dst	*	*	*	*
ADDC(.B)	src,dst	src + dst + C -> dst	*	*	*	*
SUB(.B)	src,dst	dst + .not.src + 1 -> dst	*	*	*	*
SUBC(.B)	src,dst	dst + .not.src + C -> dst	*	*	*	*
CMP(.B)	src,dst	dst - src	*	*	*	*
DADD(.B)	src,dst	src + dst + C -> dst (десятичное)	*	*	*	*
BIT(.B)	src,dst	src .and. dst	0	*	*	*
BIC(.B)	src,dst	.not.src .and. dst -> dst	-	-	-	-
BIS(.B)	src,dst	src .or. dst -> dst	-	-	-	-
XOR(.B)	src,dst	src .xor. dst -> dst	*	*	*	*
AND(.B)	src,dst	src .and. dst -> dst	0	*	*	*

## 11. Система команд. Команды с одиночным операндом.

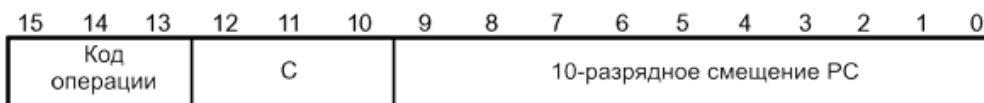
На рис.3.10 показана структура формата команды с одним операндом.



Мнемоника	S-Reg, D-Reg	Операция	Биты статуса			
			V	N	Z	C
RRC(.B)	dst	C -> MSB -> ... LSB -> C	*	*	*	*
RRA(.A)	dst	MSB -> MSB -> ... LSB -> C	0	*	*	*
PUSH(.B)	src	SP-2 -> SP, src -> @SP	-	-	-	-
SWPB	dst	Обмен байтами	-	-	-	-
CALL	dst	SP-2 -> SP, PC+2 -> @SP dst -> PC	-	-	-	-
RETI		TOS -> SR, SP+2 -> SP TOS -> PC, SP+2 -> SP	*	*	*	*
SXT	dst	Бит7 -> Бит8 ... .. Бит15	0	*	*	*

## 12. Система команд. Команды перехода.

На рис.3.11 показан формат команды условного перехода.



В таблице 3.13 приведено описание и перечень команд переходов.

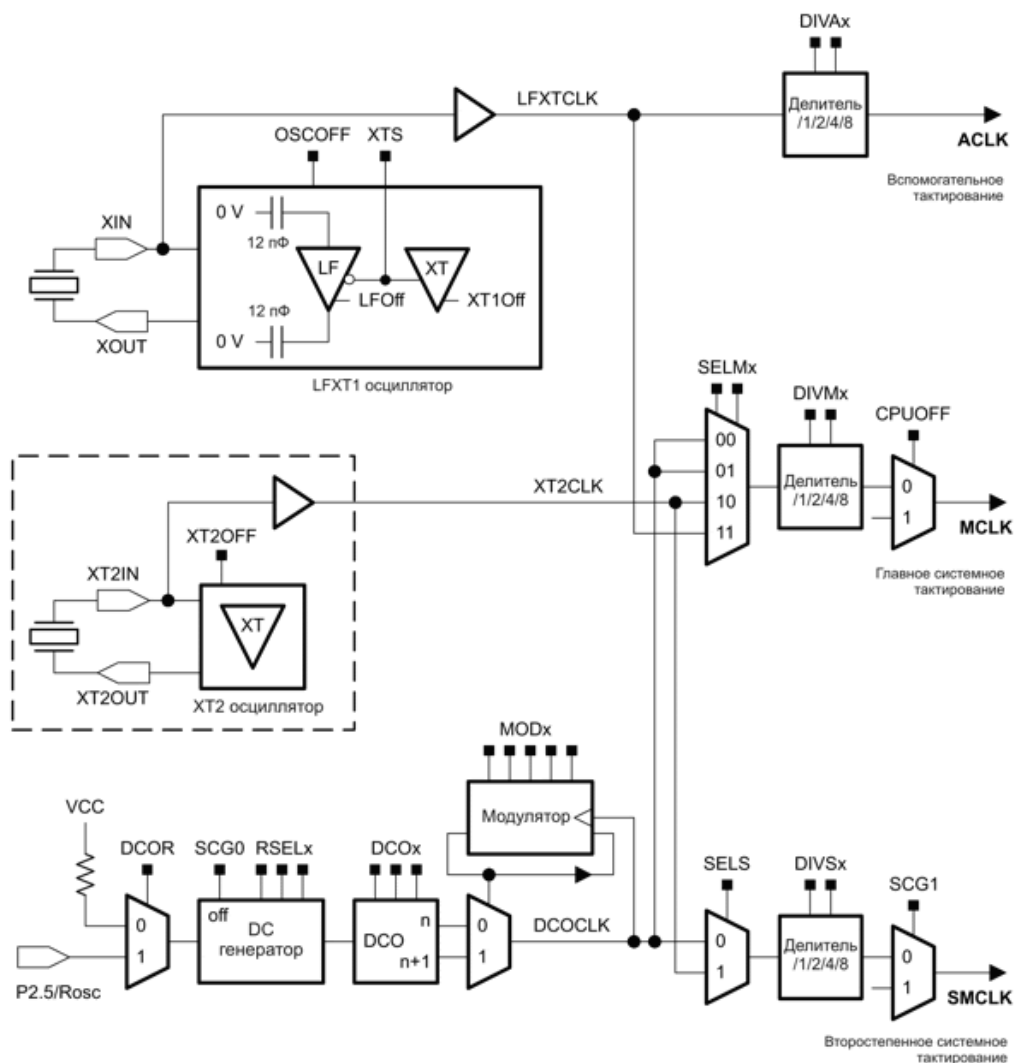
Мнемоника	S-Reg, D-Reg	Операция
JEQ/JZ	Метка	Переход к метке, если бит нуля (Z) установлен
JNE/JNZ	Метка	Переход к метке, если бит нуля (Z) сброшен
JC	Метка	Переход к метке, если бит переноса (C) установлен
JNC	Метка	Переход к метке, если бит переноса (C) сброшен
JN	Метка	Переход к метке, если бит отрицательного результата (N) установлен
JGE	Метка	Переход к метке, если (N.XOR.V)=0
JL	Метка	Переход к метке, если (N.XOR.V)=1
JMP	Метка	Безусловный переход к метке

Команды условного перехода осуществляют относительный переход по заданному смещению и не влияют на биты состояния ЦПУ. Переход может осуществляться в пределах от -511 до +512 слов относительно текущего значения PC. Величина смещения интерпретируется как 10битное значение со знаком, которое удваивается и прибавляется к содержимому счётчика команд:  $PC_{нов} = PC_{стар} + 2 + PC_{смещ} \cdot 2$ .

## 13. Основной модуль тактирования. Блок-схема.



Основной модуль тактирования обеспечивает тактирование устройств семейства MSP430x1xx. В этом разделе описывается работа с основным модулем тактирования. Этот модуль реализован во всех устройствах семейства MSP430x1xx. Имеет два или три источника тактовых импульсов: LFXT1CLK - низкочастотный / высокочастотный осциллятор, XT2CLK - дополнительный высокочастотный осциллятор, DCOCLK - встроенный осциллятор с цифровым управлением (DCO). От основного модуля тактирования можно получить три тактовых сигнала: ACLK - вспомогательное тактирование, MCLK - основное тактирование, SMCLK - второстепенное тактирование.



#### 14. Функционирование основного модуля тактирования.

После сигнала PUC источником для модулей MCLK и SMCLK является DCOCLK с частотой около 800 кГц (см. параметры в справочном руководстве конкретного устройства) и LFXT1 для модуля ACLK в режиме LF.

Управляющие биты регистра статуса SCG0, SCG1, OSCOFF и CPUOFF конфигурируют рабочие режимы MSP430 и позволяют включать или отключать отдельные части основного модуля тактирования. См. раздел «Сброс, прерывания и рабочие режимы». С помощью регистров DCOCTL, BCCTL1 и BCCTL2 осуществляется конфигурирование основного модуля тактирования. Основное тактирование может конфигурироваться и реконфигурироваться программным обеспечением в любой момент времени в ходе выполнения программы.

#### 15. Регистры основного модуля тактирования.

Регистр	Обозначение	Тип регистра	Адрес	Исходное состояние
Регистр управления DCO	DCOCTL	Чтение / запись	056h	056h после PUC
Регистр 1 управления системой основного тактирования	BCSCTL1	Чтение / запись	057h	084h после PUC
Регистр 2 управления системой основного тактирования	BCSCTL2	Чтение / запись	058h	Сбрасывается после POR
Регистр 1 разрешения прерываний SFR	IE1	Чтение / запись	0000h	Сбрасывается после PUC
Регистр 1 флагов прерываний SFR	IFG1	Чтение / запись	0002h	Сбрасывается после PUC

## 16. Контроллер флэш-памяти. Функционирование флэш-памяти.

Флэш-память в MSP430 адресуется побитно, побайтно или пословно и может перепрограммироваться. Режим по умолчанию для флэш-памяти – режим чтения. В этом режиме флэш-память не может быть стерта или записана, тактовый генератор и генератор напряжения выключены – память работает подобно ПЗУ.

Флэш-память MSP430 поддерживает внутрисистемное программирование (ISP) и не нуждается в использовании дополнительного внешнего напряжения. ЦПУ может программировать собственную флэш-память. Приведенные ниже режимы записи/стирания флэш-памяти выбираются битами BLKWRT, WRT, MERAS, ERASE:

- запись байта/слова
- запись блока
- стирание сегмента
- массовое стирание (стирание всех сегментов основной памяти)
- полное стирание (стирание всех сегментов)

Чтение или запись флэш-памяти во время программирования или стирания запрещены. Если требуется выполнение программы ЦПУ в течении записи или стирания, исполняемый код должен быть помещен в ОЗУ. Любое обновление флэш может иницироваться из флэш-памяти или ОЗУ.

## 17. Стирание флэш-памяти.

После стирания бит флэш-памяти читается как «1». Можно программировать индивидуально каждый бит, меняя его значение с «1» на «0», но перепрограммирование от «0» к «1» требует выполнения цикла стирания. Сегмент – это наименьшее количество флэш-памяти, которое можно стереть. Существует три режима стирания, которые могут быть выбраны с помощью битов ERASE и MERAS в соответствии с таблицей 5-1.

Таблица 5-1. Режимы стирания

MERAS	ERASE	Режим стирания
0	1	Стирание сегмента
1	0	Массовое стирание (стирание всех сегментов основной памяти)
1	1	Стирание всей флэш-памяти (основных и информационных сегментов)

## 18. Запись во флэш-память.

**Запись** Flash памяти можно осуществлять в режимах байт/слово или блоком, состоящим из 64 байт. Режим записи памяти определяется состоянием бита BLKWRT регистра управления. Программа, осуществляющая запись побайтно/пословно, может располагаться как во Flash памяти, так и в ОЗУ. Запись блока данных может осуществляться только из подпрограммы, расположенной в ОЗУ. Разрешение записи данных во Flash памяти производится с помощью бита WRT регистра управления.

Для записи байта/слова из Flash памяти необходимо выполнить следующие действия:

- 1) Запретить все прерывания и сторожевой таймер.
- 2) Настроить тактирование Flash контроллера.
- 3) Очистить LOCK бит.
- 4) Установить бит WRT.
- 5) Записать данные по требуемому адресу с помощью словной или байтовой формы инструкции.
- 6) Сбросить бит WRT и, если требуется, разрешить прерывания, сторожевой таймер и установить LOCK бит.

Для записи байта/слова из ОЗУ необходимо выполнить следующие действия:

- 1) Запретить все прерывания и сторожевой таймер.
- 2) Дождаться, пока состояние бита BUSY не станет равно 0.
- 3) Настроить тактирование Flash-контроллера и сбросить бит LOCK
- 4) Установить бит WRT.
- 5) Записать в память слово или байт
- 6) Дождаться, пока состояние бита BUSY не станет равно 0.
- 7) Сбросить бит WRT и, если требуется разрешить прерывания, сторожевой таймер и LOCK бит.

Время записи одного слова/байта может составить несколько миллисекунд. Существенно сократить это время можно, используя блочный метод записи. Для записи блока данных в память необходимо выполнить следующие действия:

Режимы записи, задаваемые битами WRT и BLKWRT приведены в таблице 5.2.

Таблица 5-2. Режимы записи

BLKWRT	WRT	Режим записи
--------	-----	--------------

0	1	Запись байта/слова
1	1	Запись блока

Каждый из режимов записи использует последовательность собственных команд записи, но режим блочной записи позволяет выполнять запись примерно вдвое быстрее по сравнению с режимом байт/слово, поскольку генератор напряжения остается включенным до завершения записи блока. Любая команда, модифицирующая получателя может использоваться для изменения месторасположения в флэш-памяти как в режиме записи байта/слова, так и в режиме блочной записи.

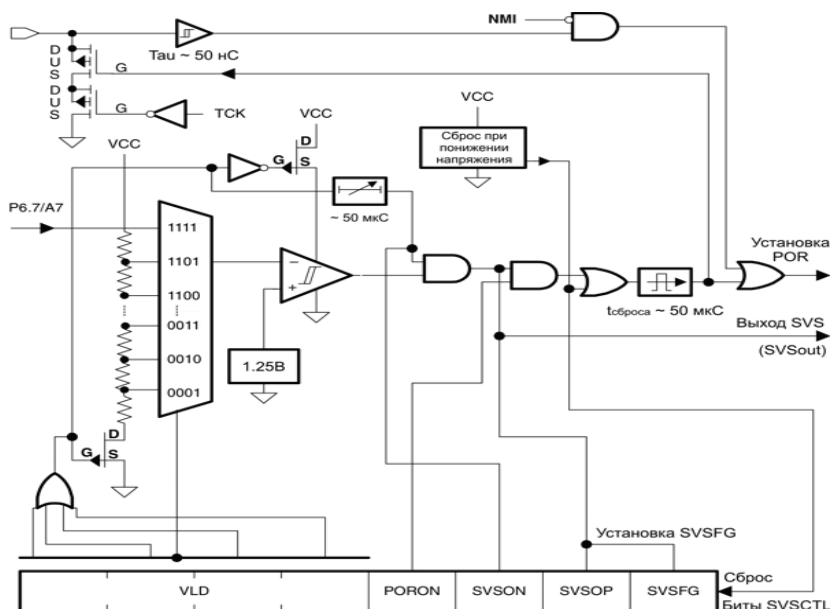
Бит BUSY установлен, пока активна процедура записи и очищается, когда запись завершена. Если операция записи инициирована из ОЗУ, ЦПУ не должен обращаться к флэш-памяти, пока BUSY=1. В противном случае произойдет нарушение прав доступа, будет установлен флаг ACCVIFG, а результат записи окажется непредсказуем.

## 19. Супервизор напряжения питания.

Супервизор напряжения питания (SVS) используется для мониторинга напряжения питания  $AV_{cc}$  или внешнего напряжения. SVS может быть сконфигурирован так, чтобы выполнялась установка флага или генерировался сигнал сброса POR, когда напряжение питания или внешнее напряжение снижаются ниже порога, установленного пользователем.

SVS обладает следующими возможностями:

- Мониторинг  $AV_{cc}$ ;
- Возможность генерации сигнала POR;
- Программно доступный вывод компаратора SVS;
- Программно доступное условие фиксации при низком напряжении;
- Выбор из 14 возможных пороговых уровней;
- Внешний канал мониторинга внешнего напряжения.



## 20. Аппаратный умножитель.

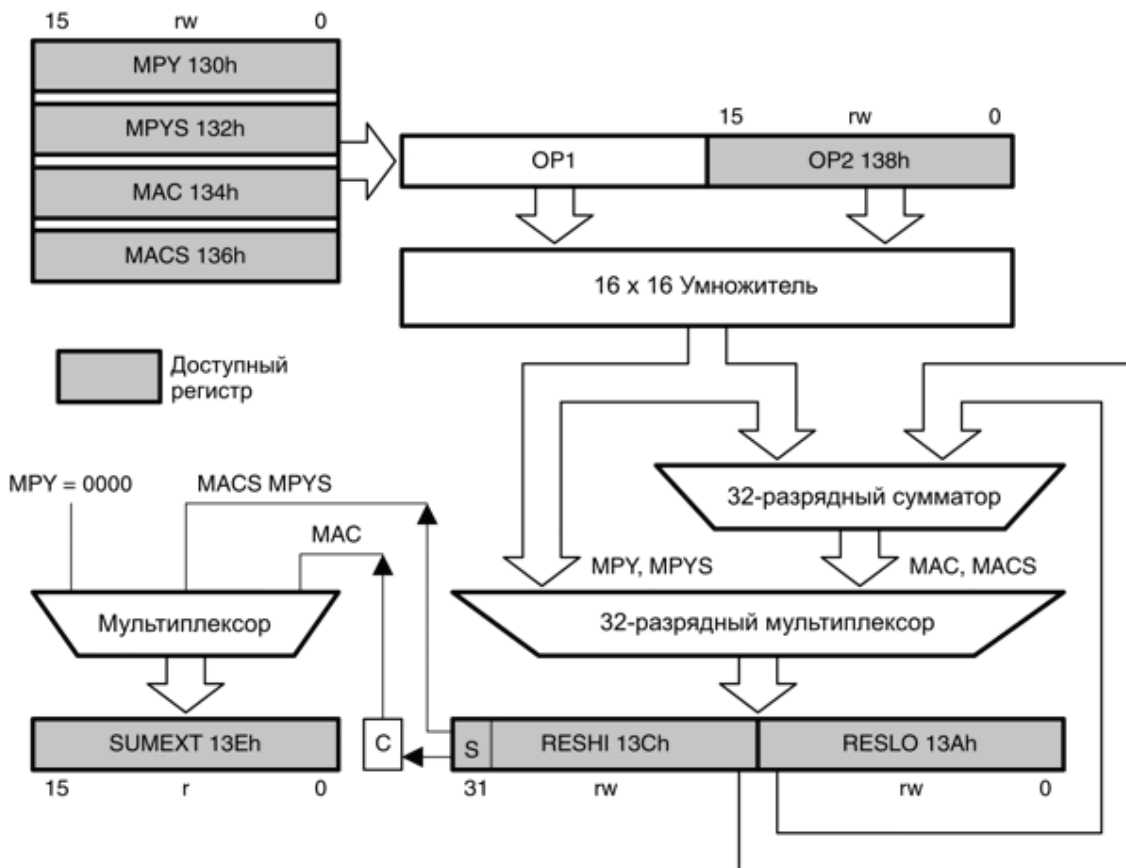
Аппаратный умножитель является периферийным устройством и не является частью ЦПУ MSP430. Это означает, что его действия не пересекаются с действиями ЦПУ. Регистры умножителя – это периферийные регистры, которые загружаются и читаются командами ЦПУ.

Аппаратный умножитель поддерживает:

- Умножение без знака;
- Умножение со знаком;
- Умножение без знака с накоплением;
- Умножение со знаком и накоплением;
- 16\*16 бит, 16\*8 бит, 8\*16 бит, 8\*8 бит.

Аппаратный умножитель имеет два 16-разрядных регистра OP1 и OP2 и три регистра результата RESLO, RESHI и SUMEXT. В регистре RESLO содержится младшее слово результата, в RESHI – старшее слово результата, а в регистре SUMEXT находится информация о результате.

Блок-схема аппаратного умножителя показана на рис.7.1.



## 21. Контроллер ПДП. Блок-схема и функционирование контроллера ПДП.

Контроллер прямого доступа к памяти (DMA) переносит данные из одного адреса в другой во всем адресном диапазоне без вмешательства ЦПУ. К примеру, контроллер DMA может переместить данные из памяти преобразования ADC12 в ОЗУ. Снижается потребляемая системой мощность, поскольку ЦПУ может оставаться в режиме пониженного энергопотребления без пробуждения при перемещении данных в/из периферии.

Контроллер DMA обладает следующими возможностями:

- Три независимых канала переноса;
- Конфигурируемые приоритеты канала DMA;
- Необходимо только два тактовых цикла MCLK;
- Возможен перенос байтов, слов или смешанно байтов/слов;
- Размер блока до 65535 байт или слов;
- Набор конфигурируемых источников запуска переноса;
- Возможность выбора условия запуска переноса по фронту/спаду или по уровню;
- Четыре режима адресации;
- Одиночный, блочный или пакетно-блочный режимы переноса.

Модуль контроллера DMA конфигурируется программным обеспечением пользователя. Контроллер DMA имеет четыре режима адресации. Режимы адресации каждого канала DMA конфигурируются независимо друг от друга. Существуют следующие режимы адресации:

- Фиксированный адрес к фиксированному адресу;
- Фиксированный адрес к блоку адресов;
- Блок адресов к фиксированному адресу;
- Блок адресов к блоку адресов.

Переносы могут быть такими: байт-байт, слово-слово, байт-слово или слово-байт.

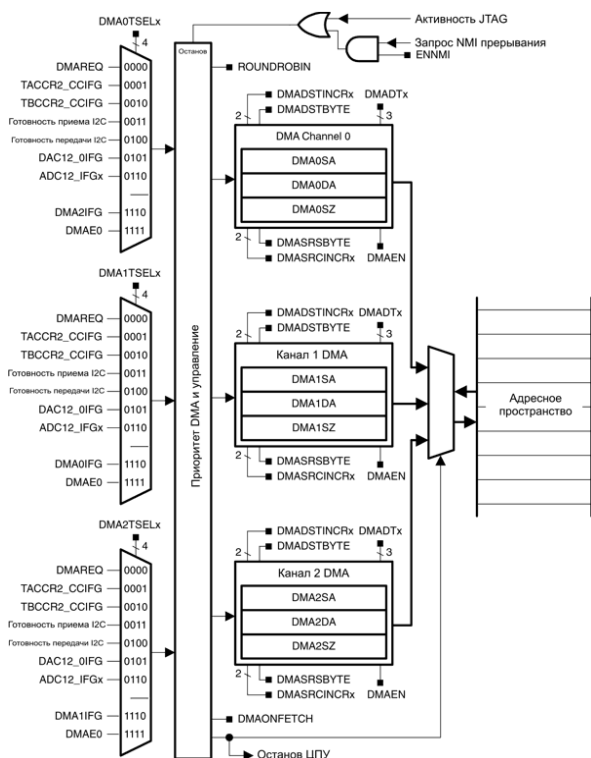


Рис.8-1. Блок-схема контроллера DMA

## 22.Режимы переноса ПДП. Одиночный перенос.

Когда канал DMA сконфигурирован в одиночном режиме переноса, пересылка каждого байта/слова требует отдельного запуска. Одиночный режим переноса устанавливается при  $DMADTx=0$ . Когда  $DMADTx=0$ , бит  $DMAEN$  очищается после каждой пересылки и должен быть установлен снова для выполнения очередной пересылки. Режим повторного одиночного переноса устанавливается при  $DMADTx=4$ . Когда  $DMATx=4$ , контроллер DMA остается включенным с  $DMAEN=1$  и DMA-перенос происходит каждый раз, когда выполняется запуск.

Регистр  $DMAxSZ$  содержит количество пересылок, которые нужно выполнить. Если  $DMAxSZ=0$ , перенос не выполняется. Содержимое регистра  $DMAxSZ$  копируется во временный регистр и уменьшается с каждым переносом. Когда регистр  $DMAxSZ$  декрементируется до нуля, он перезагружается из временного регистра и выполняется установка соответствующего флага  $DMAIFG$ .

## 23.Режимы переноса ПДП. Блочные переносы.

Блочный перенос В блочном режиме перенос полного блока данных выполняется после всего лишь одного запуска. Режим блочного переноса устанавливается при  $DMADTx=1$ . Когда  $DMADTx=1$ , бит  $DMAEN$  очищается после завершения переноса блока и должен быть установлен снова до

запуска процесса переноса другого блока. После того, как поблочный перенос был запущен, последующие сигналы запуска, появляющиеся во время переноса, игнорируются.

Режим повторного поблочного переноса устанавливается при  $DMADTx=5$ . Когда  $DMADTx=5$ , бит  $DMAEN$  остается установленным после завершения переноса блока. Следующий запуск после завершения повторного поблочного переноса запустит другой блочный перенос.

## **24. Режимы переноса ПДП. Пакетно-блочные переносы.**

Пакетно-блочный перенос – это поблочные пересылки, чередуемые с работой ЦПУ. В пакетно-блочном режиме ЦПУ выполняет 2 MCLK цикла после пересылки каждого четырех байт/слов блока. Во время пакетно-блочного переноса производительность ЦПУ составляет 20% от номинальной. Установка пакетно-блочного режима происходит при  $DMADT=\{2,3\}$ . После пакетно-блочного переноса ЦПУ возвращается к 100% производительности, а бит  $DMAEN$  очищается.  $DMAEN$  должен быть установлен снова перед запуском другой пакетно-блочной передачи. После того, как пакетно-блочный перенос был запущен, последующие сигналы запуска, появляющиеся во время пакетно-блочного переноса, игнорируются.

Повторяющийся пакетно-блочный режим устанавливается при  $DMADT=\{6,7\}$ . Когда  $DMADTx=\{6,7\}$ , бит  $DMAEN$  остается установленным после завершения пакетно-блочного переноса, и никакие дальнейшие сигналы запуска для инициирования другого пакетно-блочного переноса не требуются. Другой пакетно-блочный перенос начинается немедленно после завершения текущего пакетно-блочного переноса. В этом случае DMA-переносы должны останавливаться либо путем очистки бита  $DMAEN$ , либо по NMI-прерыванию, когда бит  $ENNM1$  установлен. В повторяющемся пакетно-блочном режиме производительность ЦПУ постоянно составляет 20%, пока повторяющийся пакетно-блочный перенос не будет остановлен.

## **25. Инициирование ПДП. Запуск по фронту, по уровню. Останов выполнения команд на время ПДП.**

Источники запуска переноса в каждом канале DMA конфигурируются независимо с помощью битов  $DMAxTSELx$ . Биты  $DMAxTSELx$  должны модифицироваться только тогда, когда бит  $DMAEN$   $DMACTLx$  равен 0.

### **Запуск по фронту**

Когда  $DMALEVEL=0$ , запуск производится по перепаду сигнала: фронт сигнала запуска иницирует пересылку. В режиме одиночного перемещения каждый DMA-перенос нуждается в собственном сигнале запуска. Когда используется блочный или пакетно-блочный режимы, для инициирования переноса требуется только один сигнал запуска.

### **Запуск по уровню**

Когда  $DMALEVEL=1$ , активизируется запуск по уровню. Правильная работа в режиме запуска по уровню возможна при выборе внешнего триггера (сигнала запуска)  $DMAE0$  в качестве DMA-триггера.

Когда  $DMALEVEL=1$ , DMA-перенос данных выполняется до тех пор, пока сигнал запуска имеет высокий уровень, а бит  $DMAEN$  остается установленным. Когда  $DMALEVEL=1$ , рекомендуется использовать режимы DMA-пересылки  $DMADTx=\{0, 1, 2, 3\}$ , поскольку бит  $DMAEN$  автоматически сбрасывается после конфигурирования DMA-переноса.

Когда  $DMALEVEL=1$ , сигнал запуска должен иметь высокий уровень при блочном или пакетно-блочном переносе до завершения переноса. Если сигнал запуска становится низким во время блочного или пакетно-блочного переноса, контроллер DMA удерживается в текущем состоянии до тех пор, пока сигнал запуска снова не станет высоким или пока не произойдет программной модификации DMA-регистров. Если DMA-регистры не модифицировались программой когда сигнал запуска опять стал высоким, перенос возобновляется с момента, когда сигнал запуска низким.

## Останов выполнения команд при выполнении DMA-переноса

Бит DMAONFETCH управляет приостановкой ЦПУ для выполнения DMA-переноса. Когда DMAONFETCH=0, при получении сигнала DMA-запуска ЦПУ немедленно останавливается и начинается DMA-перенос. Когда DMAONFETCH=1, ЦПУ завершает текущую выполняемую команду, после чего контроллер DMA останавливает ЦПУ и начинает DMA-перенос.

## 26.Регистры контроллера ПДП.

### Регистры DMA

Перечень регистров DMA приведен в таблице 8.3.

Таблица 8-3. Регистры DMA

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Регистр 0 управления DMA	DMACTL0	Чтение/запись	0122h	Сбрасывается с POR
Регистр 1 управления DMA	DMACTL1	Чтение/запись	0124h	Сбрасывается с POR
Регистр управления канала 0 DMA	DMA0CTL	Чтение/запись	01E0h	Сбрасывается с POR
Регистр адреса источника канала 0 DMA	DMA0SA	Чтение/запись	01E2h	Не изменяется
Регистр адреса получателя канала 0 DMA	DMA0DA	Чтение/запись	01E4h	Не изменяется
Регистр объема переноса канала 0 DMA	DMA0SZ	Чтение/запись	01E6h	Не изменяется
Регистр управления канала 1 DMA	DMA1CTL	Чтение/запись	01E8h	Сбрасывается с POR
Регистр адреса источника канала 1 DMA	DMA1SA	Чтение/запись	01EAh	Не изменяется
Регистр адреса получателя канала 1 DMA	DMA1DA	Чтение/запись	01ECh	Не изменяется
Регистр объема переноса канала 1 DMA	DMA1SZ	Чтение/запись	01EEh	Не изменяется
Регистр управления канала 2 DMA	DMA2CTL	Чтение/запись	01F0h	Сбрасывается с POR
Регистр адреса источника канала 2 DMA	DMA2SA	Чтение/запись	01F2h	Не изменяется
Регистр адреса получателя канала 2 DMA	DMA2DA	Чтение/запись	01F4h	Не изменяется
Регистр объема переноса канала 2 DMA	DMA2SZ	Чтение/запись	01F6h	Не изменяется

## 27.Цифровые порты ввода/вывода.



Устройства MSP430 имеют до 6 портов цифровых входов/выходов от P1 до P6. Каждый порт имеет 8 выводов входа/выхода. Каждый вывод индивидуально конфигурируется как вход или выход и каждая линия ввода/вывода может быть индивидуально считана или записана.

Порты P1 и P2 имеют возможность вызывать прерывание. Для каждой линии ввода/вывода портов P1 и P2 можно индивидуально разрешить прерывания и сконфигурировать их так, чтобы прерывание происходило по фронту или спаду входного сигнала. Все линии ввода/вывода порта P1 являются источником одного вектора прерывания, а все линии ввода/вывода порта P2 – источник другого вектора прерывания.

Цифровые входы/выходы обладают следующими возможностями:

- Независимые индивидуально программируемые входы/выходы;
- Любые комбинации входа или выхода;
- Индивидуально конфигурируемые прерывания от P1 и P2;
- Раздельные регистры данных для входов и выходов.

Регистры:

Порт	Регистр	Краткое обозначение	Адрес	Тип регистра	Исходное состояние
P2	Ввод	P2IN	028h	Только чтение	—
	Вывод	P2OUT	029h	Чтение/запись	Не изменяется
	Направление	P2DIR	02Ah	Чтение/запись	Сброс с PUC
	Флаг прерывания	P2IFG	02Bh	Чтение/запись	Сброс с PUC
	Выбор фронта прерывания	P2IES	02Ch	Чтение/запись	Не изменяется
	Разрешение прерывания	P2IE	02Dh	Чтение/запись	Сброс с PUC
	Выбор порта	P2SEL	02Eh	Чтение/запись	Сброс с PUC
P3	Ввод	P3IN	018h	Только чтение	—
	Вывод	P3OUT	019h	Чтение/запись	Не изменяется
	Направление	P3DIR	01Ah	Чтение/запись	Сброс с PUC
	Выбор порта	P3SEL	01Bh	Чтение/запись	Сброс с PUC

## 28. Сторожевой таймер.

Первичная функция модуля сторожевого таймера (WDT) – выполнять рестарт управляемой системы при возникновении проблемы с программным обеспечением. Если установленный временной интервал истек, генерируется системный сброс. Если сторожевая функция в приложении не нужна, модуль может быть сконфигурирован как интервальный таймер для генерации прерываний через установленные интервалы времени.

Сторожевой таймер обладает следующими возможностями:

- Восемь программно настраиваемых временных интервалов
- Режим сторожевого таймера
- Режим интервального отсчета
- Доступ к регистру управления WDT защищен паролем
- Управление функцией вывода nonRST/NMI
- Возможность выбора источника тактовых импульсов
- Возможность останова для уменьшения потребляемой мощности

Блок схема модуля WDT показана на рис.9-1.

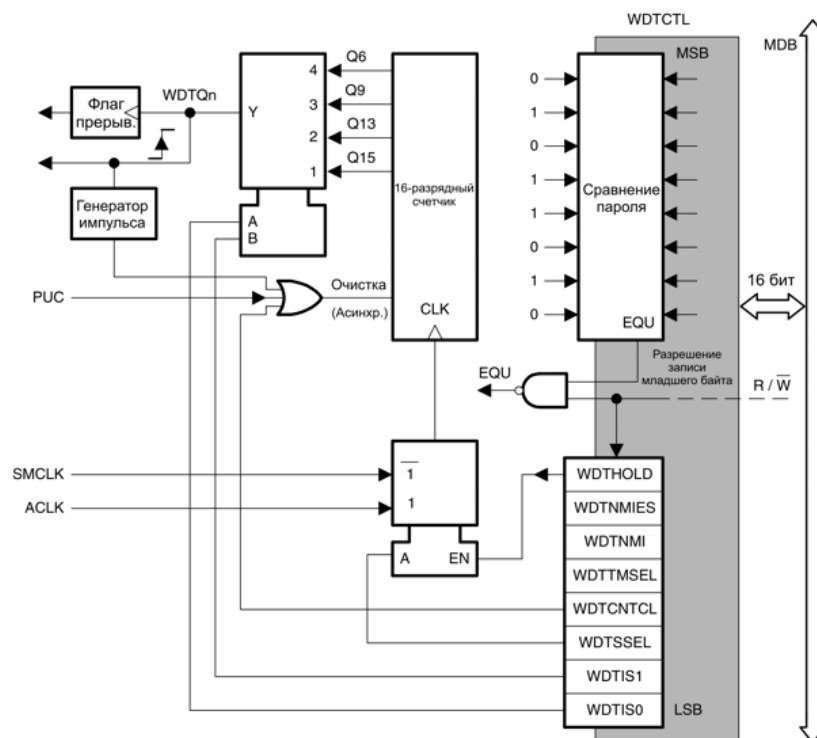


Рис.10-1 Блок-схема сторожевого таймера

## 29. Таймер А. Структура и режимы работы.

Таймер А – это 16-разрядный таймер/счетчик с тремя регистрами захвата/сравнения. Таймер А может обеспечить множество захватов/сравнений, выходов ШИМ и выдержку временных интервалов. Таймер А также имеет обширные возможности прерывания. Прерывания могут быть сгенерированы от счетчика при переполнении и от каждого из регистров захвата/сравнения.

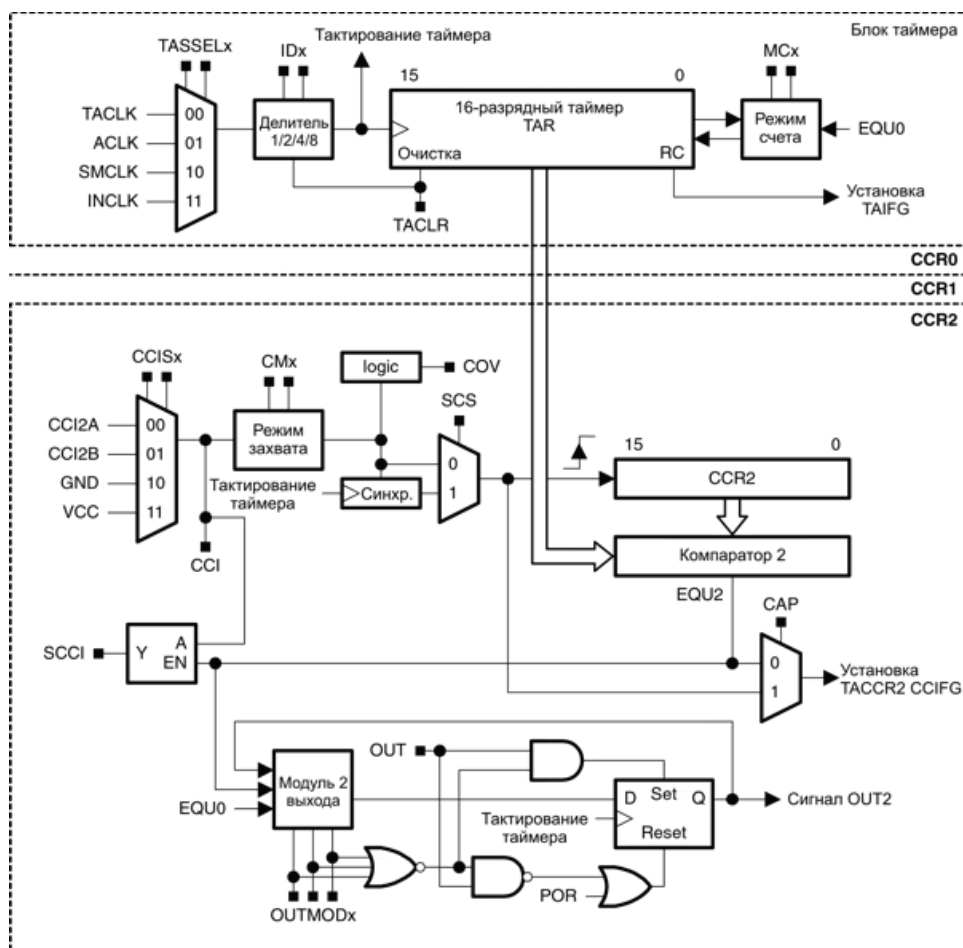


Рис.11-1 Блок-схема таймера А

Таймер имеет четыре режима работы, описанных в таблице 11-1: «стоп», «вверх», «непрерывный» и «вверх/вниз». Режимы работы выбираются с помощью битов MCx.

Таблица 11-1. Режимы таймера.

MCx	Режим	Описание
00	Стоп	Останов таймера
01	Вверх	Таймер многократно считает от нуля до значения в TACCR0
10	Непрерывный	Таймер многократно считает от нуля до значения 0FFFFh
11	Вверх/вниз	Таймер многократно считает от нуля вверх до значения в TACCR0 и назад до нуля.

### 30. Таймер А. Модуль вывода.

Каждый блок захвата/сравнения содержит модуль вывода. Модуль вывода используется для генерации выходных сигналов, в т.ч. таких, как ШИМ-сигналы. Каждый модуль вывода имеет восемь рабочих режимов, которые генерируют сигналы, основываясь на сигналах EQU0 и EQUx.

Режимы вывода устанавливаются битами OUTMODx, их описание приведено в таблице 11-2. Сигнал OUTx изменяется с нарастающим фронтом тактового сигнала таймера во всех режимах, кроме режима 0. Режимы вывода 2, 3, 6 и 7 не используются для модуля вывода 0, потому что

EQUx=EQU0.

Таблица 11-2. Режимы вывода.

OUTMODx	Режим	Описание
000	Вывод	Выходной сигнал OUTx определяется битом OUTx. Сигнал OUTx изменяется немедленно при изменении OUTx.
001	Установка	Выход устанавливается, когда таймер досчитывает до значения в TACCRx. Он остается установленным до сброса таймера или до выбора другого режима вывода и воздействия на выход.
010	Переключение/Сброс	Выход переключается, когда таймер досчитывает до значения TACCRx. Он сбрасывается, когда таймер досчитывает до значения TACCR0.
011	Установка/Сброс	Выход устанавливается, когда таймер досчитывает до значения TACCRx. Он сбрасывается, когда таймер досчитывает до значения TACCR0.
100	Переключение	Выход переключается, когда таймер досчитывает до значения TACCRx. Период выходного сигнала равен удвоенному периоду таймера.
101	Сброс	Выход сбрасывается, когда таймер досчитывает до значения TACCRx. Это остается сброшенным до выбора другого режима вывода и воздействия на выход.
110	Переключение/Установка	Выход переключается, когда таймер досчитывает до значения TACCRx. Он устанавливается, когда таймер досчитывает до значения TACCR0.
111	Сброс/Установка	Выход сбрасывается, когда таймер досчитывает до значения TACCRx. Он устанавливается, когда таймер досчитывает до значения TACCR0.

### 31.Регистры Таймера А.

Перечень регистров Таймера А приведен в таблице 11-3.

Таблица 11-1. Регистры Таймера А.

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Управление Таймером А	TACTL	Чтение/запись	0160h	Сброс с POR
Счетчик Таймера А	TAR	Чтение/запись	0170h	Сброс с POR
Регистр 0 управления захватом/сравнением Таймера А	TACCTL0	Чтение/запись	0162h	Сброс с POR
Регистр 0 захвата/сравнения Таймера А	TACCR0	Чтение/запись	0172h	Сброс с POR
Регистр 1 управления захватом/сравнением Таймера А	TACCTL1	Чтение/запись	0164h	Сброс с POR
Регистр 1 захвата/сравнения Таймера А	TACCR1	Чтение/запись	0174h	Сброс с POR

Регистр 2 управления захватом/сравнением Таймера А	TACCTL2	Чтение/запись	0166h	Сброс с POR
Регистр 2 захвата/сравнения Таймера А	TACCR2	Чтение/запись	0176h	Сброс с POR
Вектор прерывания Таймера А	TAIV	Только чтение	012Eh	Сброс с POR

### 32. Таймер В. Структура и режимы работы. (В ПАЧКЕ С СИСТ. АНАЛИЗОМ!!!)

Таймер В – это 16-разрядный таймер/счетчик с тремя или семью регистрами захвата/сравнения. Таймер В может поддерживать несколько режимов захвата/сравнения, вывод ШИМ-сигналов и выдержку временных интервалов. Таймер В также имеет расширенные возможности прерываний. Прерывания могут быть сгенерированы при переполнении счетчика и от каждого из регистров захвата/сравнения.

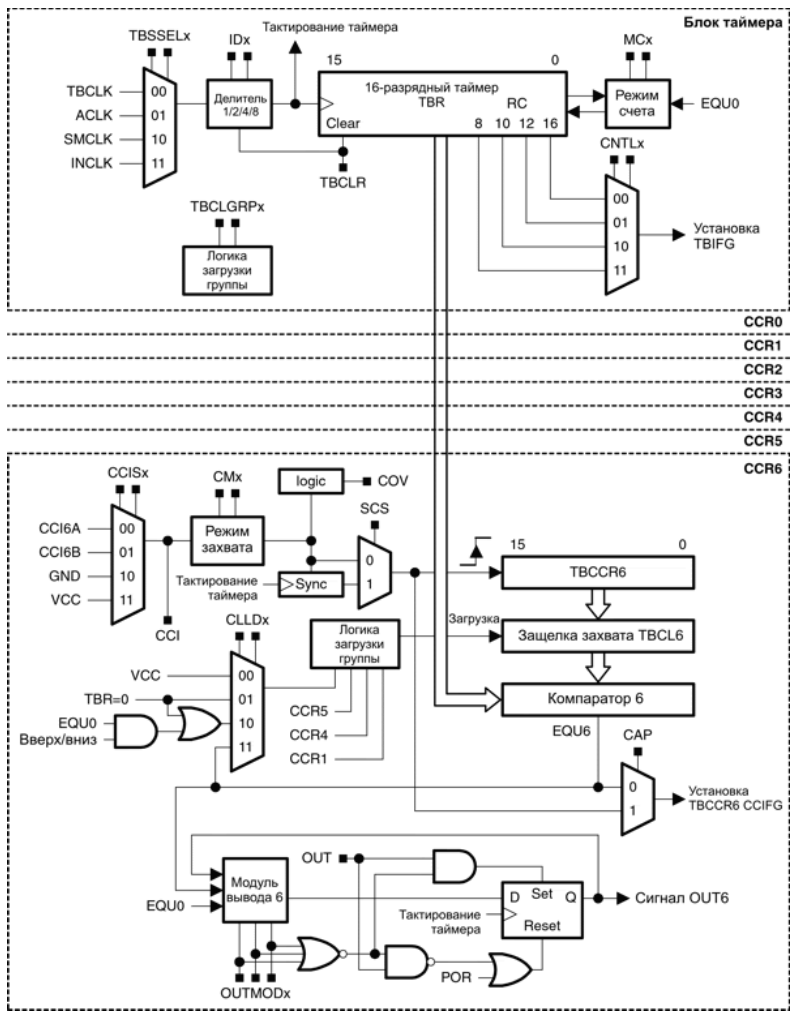


Рис.12-1 Блок-схема таймера В

Таймер имеет четыре режима работы, описанных в таблице 12-1: «стоп», «вверх», «непрерывный» и «вверх/вниз». Рабочий режим выбирается с помощью битов MCx.

Таблица 12-1. Режимы таймера.

МСх	Режим	Описание
00	Стоп	Останов таймера
01	Вверх	Таймер многократно считает от нуля до значения в регистре сравнения <b>TBCL0</b>
10	Непрерывный	Таймер многократно считает от нуля до значения, выбранного битами TBCTLx.
11	Вверх/вниз	Таймер многократно считает от нуля вверх до значения в <b>TBCL0</b> и назад до нуля.

### 33. Таймер В. Модуль вывода.

Таймер В – это 16-разрядный таймер/счетчик с тремя или семью регистрами захвата/сравнения. Таймер В может поддерживать несколько режимов захвата/сравнения, вывод ШИМ-сигналов и выдержку временных интервалов. Таймер В также имеет расширенные возможности прерываний. Прерывания могут быть сгенерированы при переполнении счетчика и от каждого из регистров захвата/сравнения.

Каждый блок захвата/сравнения содержит модуль вывода. Модуль вывода используется для генерации выходных сигналов, таких как ШИМ-сигналы. Каждый модуль вывода имеет восемь рабочих режимов, которые генерируют сигналы, основываясь на сигналах EQU0 и EQUx. Функция ножки TBoutH может использоваться для установки всех выходов таймера В в «третье» (высокоимпедансное) состояние. Когда для ножки выбрана функция TBoutH и на вывод подан высокий лог. уровень, все выходы таймера В находятся в «третьем» состоянии.

#### Режимы вывода

Режимы вывода задаются битами OUTMODx, а их описание приведено в таблице 12-4. Сигнал OUTx изменяется по нарастающему фронту тактового сигнала таймера во всех режимах, кроме режима 0. Режимы вывода 2, 3, 6 и 7 не используются для модуля вывода 0, поскольку EQUx=EQU0.

Таблица 12-4. Режимы вывода.

OUTMODx	Режим	Описание
000	Вывод	Выходной сигнал OUTx определяется битом OUTx. Сигнал OUTx изменяется немедленно при изменении OUTx.
001	Установка	Выход устанавливается, когда таймер досчитывает до значения в TBCLx. Он остается установленным до сброса таймера или до выбора другого режима вывода и воздействия на выход.
010	Переключение/сброс	Выход переключается, когда таймер досчитывает до значения TBCLx. Он сбрасывается, когда таймер досчитывает до значения TBCL0.
011	Установка/сброс	Выход устанавливается, когда таймер досчитывает до значения TBCLx. Он сбрасывается, когда таймер досчитывает до значения TBCL0.
100	Переключение	Выход переключается, когда таймер досчитывает до значения TBCLx. Период выходного сигнала равен удвоенному периоду таймера.
101	Сброс	Выход сбрасывается, когда таймер досчитывает до значения TBCLx. Он остается сброшенным до выбора другого режима вывода и воздействия на выход.
110	Переключение/установка	Выход переключается, когда таймер досчитывает до

		значения TBCLx. Он устанавливается, когда таймер досчитывает до значения TBCL0.
111	Сброс/установка	Выход сбрасывается, когда таймер досчитывает до значения TBCLx. Он устанавливается, когда таймер досчитывает до значения TBCL0.

### 34.Регистры Таймера В.

Перечень регистров таймера В приведен в таблице 12-5.

Таблица 12-5. Регистры Таймера В.

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Управление Таймером В	TBCTL	Чтение/запись	0180h	Сброс с POR
Счетчик Таймера В	TBR	Чтение/запись	0190h	Сброс с POR
Регистр 0 управления захватом/сравнением таймера В	TBCCTL0	Чтение/запись	0182h	Сброс с POR
Регистр 0 захвата/сравнения таймера В	TBCCR0	Чтение/запись	0192h	Сброс с POR
Регистр 1 управления захватом/сравнением таймера В	TBCCTL1	Чтение/запись	0184h	Сброс с POR
Регистр 1 захвата/сравнения таймера В	TBCCR1	Чтение/запись	0194h	Сброс с POR
Регистр 2 управления захватом/сравнением таймера В	TBCCTL2	Чтение/запись	0186h	Сброс с POR
Регистр 2 захвата/сравнения таймера В	TBCCR2	Чтение/запись	0196h	Сброс с POR
Регистр 3 управления захватом/сравнением таймера В	TBCCTL3	Чтение/запись	0188h	Сброс с POR
Регистр 3 захвата/сравнения таймера В	TBCCR3	Чтение/запись	0198h	Сброс с POR
Регистр 4 управления захватом/сравнением таймера В	TBCCTL4	Чтение/запись	018Ah	Сброс с POR
Регистр 4 захвата/сравнения таймера В	TBCCR4	Чтение/запись	019Ah	Сброс с POR
Регистр 5 управления захватом/сравнением таймера В	TBCCTL5	Чтение/запись	018Ch	Сброс с POR
Регистр 5 захвата/сравнения таймера В	TBCCR5	Чтение/запись	019Ch	Сброс с POR
Регистр 6 управления захватом/сравнением таймера В	TBCCTL6	Чтение/запись	018Eh	Сброс с POR
Регистр 6 захвата/сравнения таймера В	TBCCR6	Чтение/запись	019Eh	Сброс с POR
Вектор прерывания Таймера В	TBIV	Только чтение	011Eh	Сброс с POR

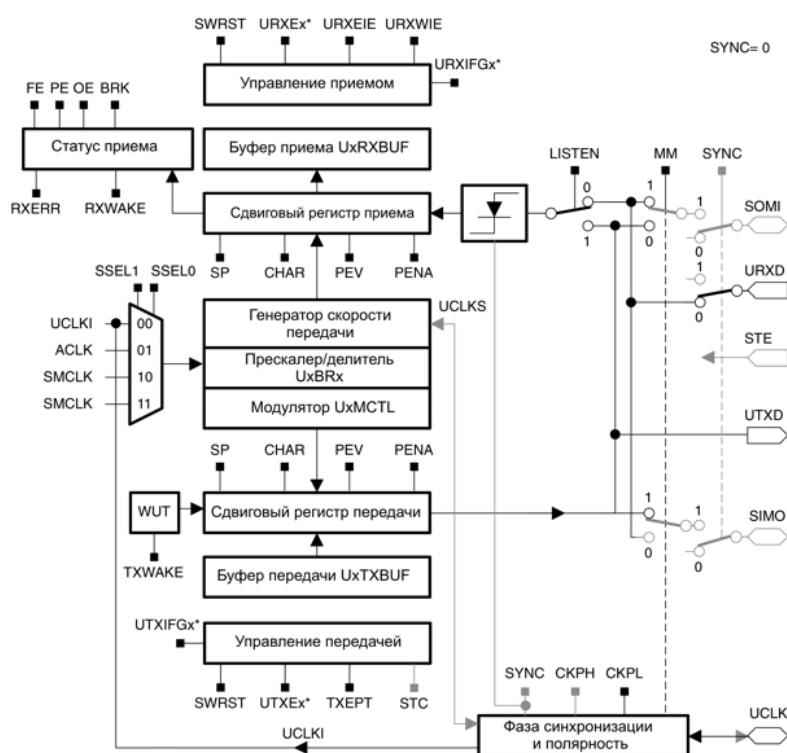
### 35.Периферийный интерфейс USART, режим UART. Инициализация и сброс USART.

Универсальный синхронно/асинхронный приемопередающий (USART) периферийный интерфейс поддерживает два последовательных режима в одном аппаратном модуле. Этот раздел описывает работу асинхронного режима UART. USART0 реализован в устройствах MSP430x12xx, MSP430x13xx и MSP430x15x. В дополнение к USART0 в устройствах MSP430x14x и MSP430x16x реализован второй идентичный USART модуль – USART1.

В режиме UART модуль USART передает и принимает символы на скорости, асинхронной другому устройству. Синхронизация каждого символа основана на выбранной скорости передачи USART. Для выполнения функций передачи и приема используется одинаковая скорость в бодах.

Модуль USART сбрасывается сигналом PUC или при установке бита SWRST. После PUC бит SWRST автоматически устанавливается, оставляя USART в состоянии сброса. Когда бит SWRST установлен, сброшены биты URXIE<sub>x</sub>, UTXIE<sub>x</sub>, URXIFG<sub>x</sub>, RXWAKE, TXWAKE, RXERR, BRK, PE, OE, FE и установлены биты UTXIFG<sub>x</sub> и TXEPT. Флаги разрешения приема и передачи URXEx и UTXEx не изменяются битом SWRST. Очистка SWRST позволяет модулю USART функционировать. См. также раздел «Модуль USART, режим I2C» при реконфигурировании USART0 из режима I2C в режим UART.

На рис.13-1 показан USART, сконфигурированный в режиме UART.



\*См. справочное руководство конкретного устройства для выяснения расположения SFR

Рис.13-1 Блок-схема USART в режиме UART

### 36.Периферийный интерфейс USART, режим UART. Асинхронные коммуникационные форматы.

Когда два устройства обмениваются информацией асинхронно, в качестве протокола используется формат «свободная линия». Когда связываются три или более устройств, USART поддерживает многопроцессорные коммуникационные форматы со свободной линией и формат с адресным битом.



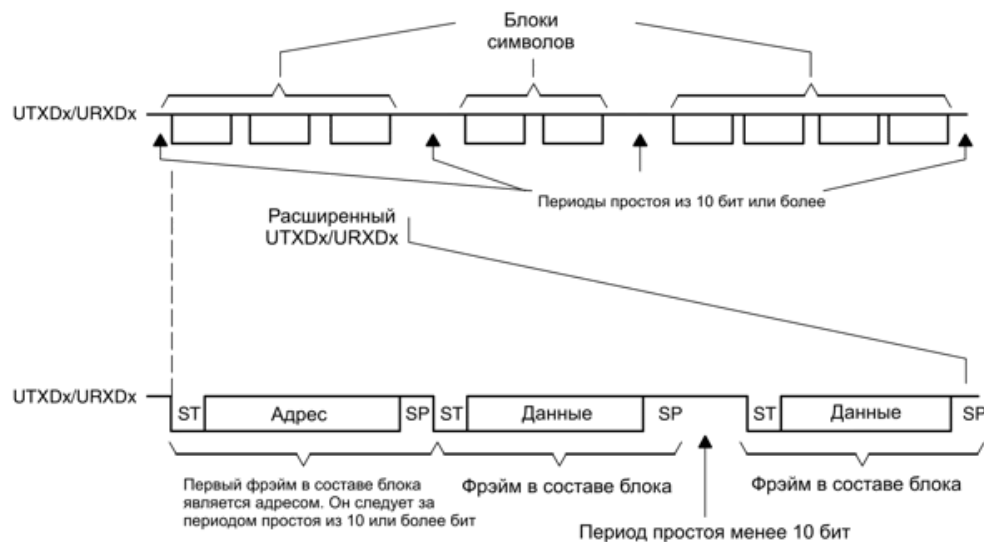


Рис.13-3 Формат свободной линии

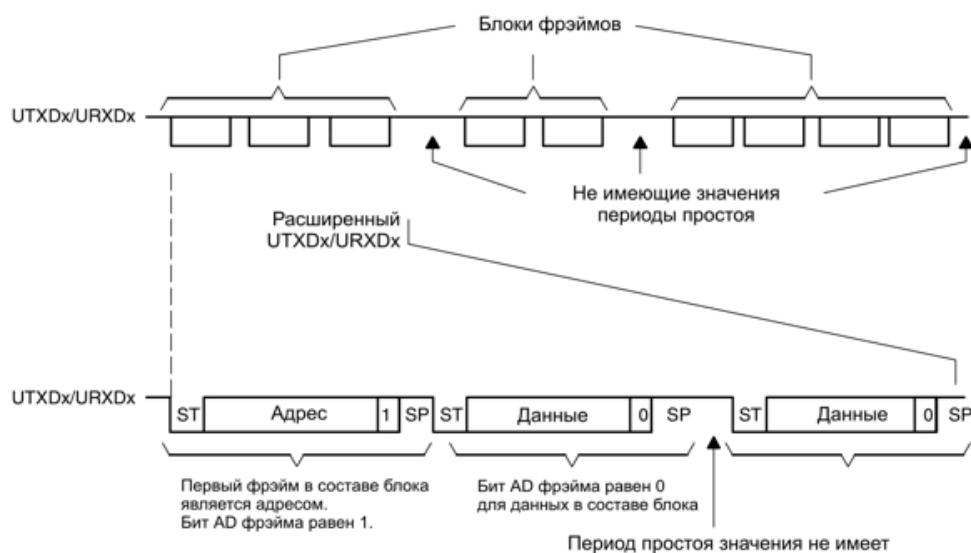


Рис.13-4 Многопроцессорный формат с адресным битом

### 37. Контроллер скорости передачи UART.

Контроллер (генератор) скорости передачи USART может создавать стандартные скорости передачи от источников нестандартных частот. Контроллер скорости передачи использует один прескалер/делитель и модулятор, показанные на рис.13-7. Эта комбинация позволяет получить дробные коэффициенты деления при генерации скорости передачи в бодах. Максимальная скорость передачи USART составляет одну треть источника таковой частоты USART BRCLK.

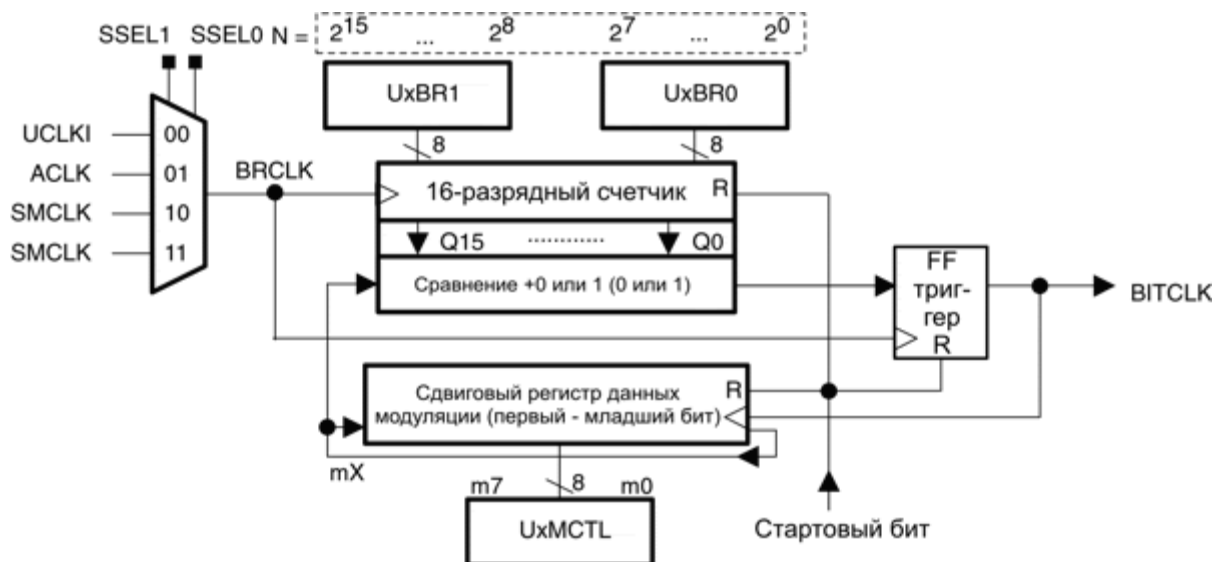


Рис.13-7 Контроллер генератора передачи MSP430

### 38.Прерывания UART.

#### Функционирование прерывания USART при передаче

Флаг прерывания UTXIFGx устанавливается передатчиком для индикации готовности UxTXBUF к приему другого символа. Запрос прерывания генерируется, если установлены флаги UTXIEx и GIE. UTXIFGx автоматически сбрасывается, если запрос прерывания обслужен или если символ записан в UxTXBUF.

UTXIFGx устанавливается после PUC или когда SWRST=1. UTXIEx сбрасывается после PUC или когда SWRST=1. Это показано на рис.13-10.

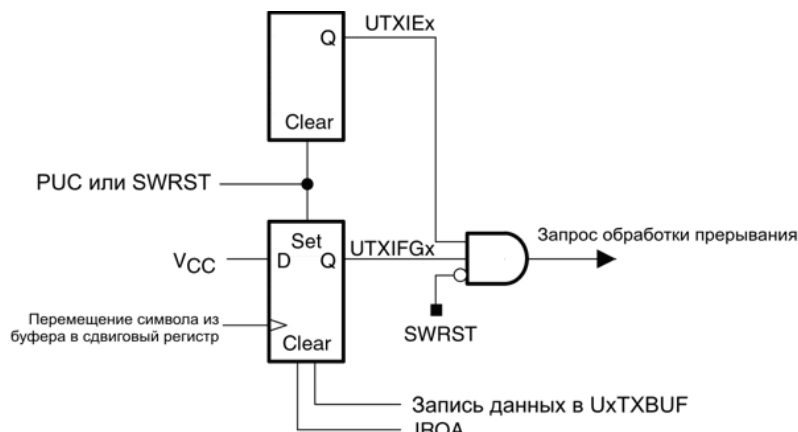


Рис.13-10. Прерывание при передаче

#### Функционирование прерывания USART при приеме

Флаг прерывания URXIFGx устанавливается каждый раз при приеме символа и его загрузке в UxRXBUF. Запрос прерывания генерируется, если также установлены флаги URXIEx и GIE. URXIFGx и URXIEx сбрасываются сигналом системного сброса PUC или когда SWRST=1. URXIFGx сбрасывается автоматически, если запрос прерывания обработан (когда URXSE=0) или когда прочитан UxRXBUF. Это показано на рис.13-11.

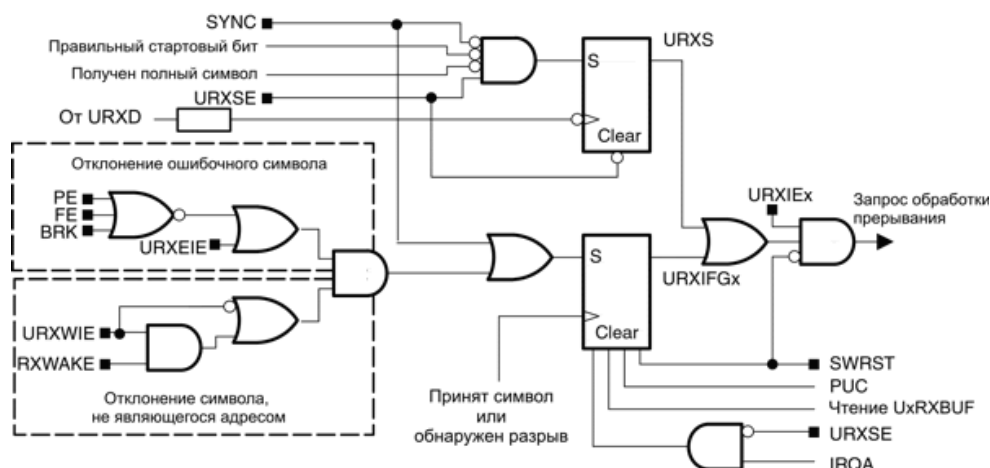


Рис.13-11. Прерывание при приеме

### 39.Регистры USART, режим UART.

В таблице 13-3 приведен перечень регистров для всех устройств с модулем USART. Таблица 13-4 справедлива только для устройств со вторым USART модулем - USART1.

Таблица 13-3. Регистры управления и статуса USART0

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Регистр управления USART	U0CTL	Чтение/запись	070h	001h после PUC
Регистр управления передачей	U0TCTL	Чтение/запись	071h	001h после PUC
Регистр управления приемом	U0RCTL	Чтение/запись	072h	000h после PUC
Регистр управления модуляцией	U0MCTL	Чтение/запись	073h	Не изменяется
Регистр 0 управления скоростью передачи	U0BR0	Чтение/запись	074h	Не изменяется
Регистр 1 управления скоростью передачи	U0BR1	Чтение/запись	075h	Не изменяется
Регистр буфера приема	U0RXBUF	Чтение	076h	Не изменяется
Регистр буфера передачи	U0TXBUF	Чтение/запись	077h	Не изменяется
Регистр 1 включения модуля SFR*	ME1	Чтение/запись	004h	000h после PUC
Регистр 1 разрешения прерывания SFR*	IE1	Чтение/запись	000h	000h после PUC
Регистр 1 флага прерывания SFR*	IFG1	Чтение/запись	002h	082h после PUC

\* Не применимо к устройствам `12xx. См. описания регистров для выяснения расположения регистров и бит у этих устройств.

Таблица 13-4. Регистры управления и статуса USART1

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Регистр управления USART	U1CTL	Чтение/запись	078h	001h после PUC
Регистр управления передачей	U1TCTL	Чтение/запись	079h	001h после PUC
Регистр управления приемом	U1RCTL	Чтение/запись	07Ah	000h после PUC
Регистр управления модуляцией	U1MCTL	Чтение/запись	07Bh	Не изменяется
Регистр 0 управления скоростью передачи	U1BR0	Чтение/запись	07Ch	Не изменяется

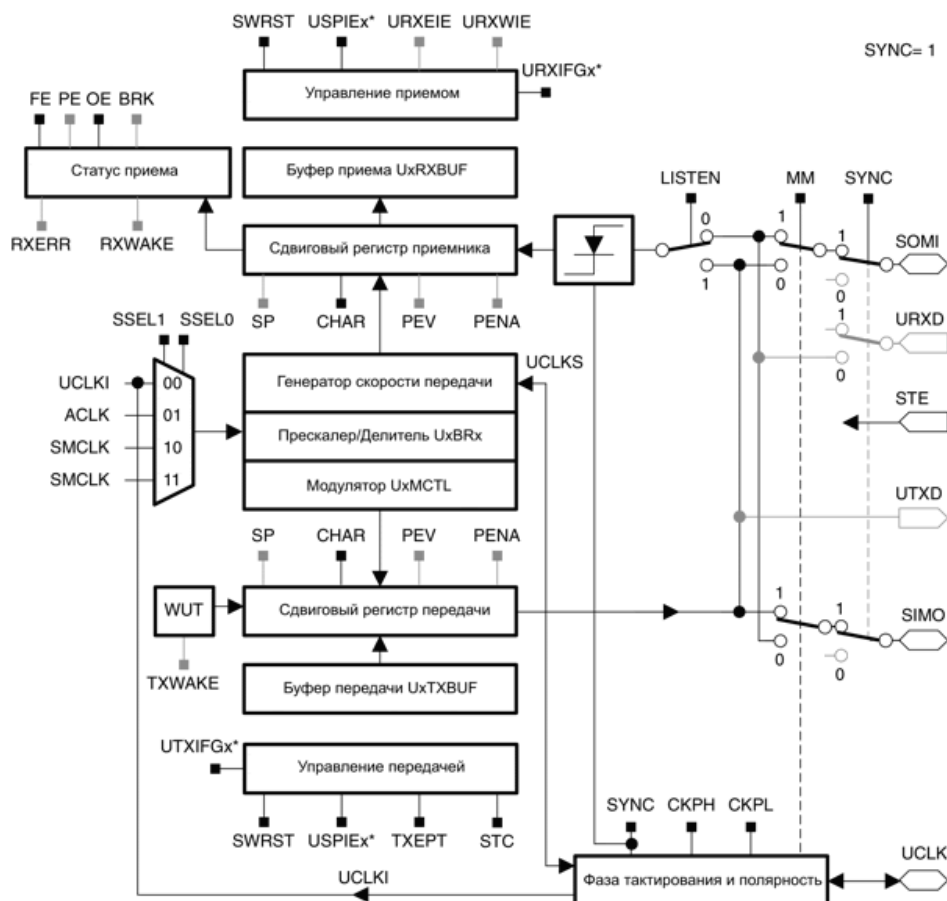
Регистр 1 управления скоростью передачи	U1BR1	Чтение/запись	07Dh	Не изменяется
Регистр буфера приема	U1RXBUF	Чтение	07Eh	Не изменяется
Регистр буфера передачи	U1TXBUF	Чтение/запись	07Fh	Не изменяется
Регистр 2 включения модуля SFR	ME2	Чтение/запись	005h	000h после PUC
Регистр 2 разрешения прерывания SFR	IE2	Чтение/запись	001h	000h после PUC
Регистр 2 флага прерывания SFR	IFG2	Чтение/запись	003h	000h после PUC

В режиме UART модуль USART передает и принимает символы на скорости, асинхронной другому устройству. Синхронизация каждого символа основана на выбранной скорости передачи USART. Для выполнения функций передачи и приема используется одинаковая скорость в бодах.

При обмене данными в асинхронном режиме UART каждый передаваемый символ содержит стартовый бит, семь или восемь битов данных, бит контроля чётности и один или два стоповых бит. Период следования битов определяется выбранным источником тактовых импульсов и настройкой работы сдвиговых регистров, преобразующих поступающий по шине данных MDB параллельный код в последовательный, а принимаемый последовательный код – в параллельный для передачи его на шину MDB. Скорость передачи определяется частотой тактирования принимающего и передающего сдвиговых регистров. В асинхронном режиме обмена данными между двумя устройствами используется протокол с форматом «свободная линия». При таком формате блоки данных на линиях передачи или приёма разделены временем простоя (временем, когда линия свободна). Простой линии приёма обнаруживается, когда приняты 10 или более не изменяющихся со временем логических единиц (меток) после первого стопового бита символа. При использовании двух стоповых битов, второй стоповый бит принимается за первый маркерный бит периода простоя. Когда связываются три или более устройств, к биту данных добавляется адресный 9 бит, что позволяет USART поддерживать многопроцессорные коммуникационные форматы со «свободной линией».

#### 40.Периферийный интерфейс USART, режим SPI. Функционирование SPI.

В синхронном режиме USART подключает MSP430 к внешней системе через три или четыре вывода: SIMO, SOMI, UCLK и STE. Режим SPI выбирается, когда бит SYNC установлен, а бит I2C очищен.



\* См. справочное руководство конкретного устройства для определения расположения SFR

Рис.14-1. Блок-схема USART в режиме SPI

В синхронном режиме последовательные данные передаются и принимаются множеством устройств с использованием общего тактирования, обеспечиваемого ведущим. Дополнительный вывод STE, управляемый ведущим, необходим для разрешения приема и передачи данных устройством.

Три или четыре сигнала используются для обмена данными через SPI:

- Вход ведомого, выход ведущего  
SIMO Режим ведущего: SIMO – линия вывода данных  
Режим ведомого: SIMO – линия ввода данных
- Выход ведомого, вход ведущего  
SOMI Режим ведущего: SOMI – линия ввода данных  
Режим ведомого: SOMI – линия вывода данных
- Тактирование USART SPI  
UCLK Режим ведущего: UCLK – выход  
Режим ведомого: UCLK – вход
- STE Разрешение передачи ведомого. Используется в 4-выводном режиме, когда на одной шине может быть много ведущих. Не применяется в 3-выводном режиме.  
4-х выводной режим ведущего:  
Когда STE имеет высокий уровень, SIMO и UCLK работают как обычно.  
Когда STE имеет низкий уровень, SIMO и UCLK устанавливаются на направление ввода.  
4-х выводной режим ведомого:

Когда STE имеет высокий уровень, функционирование RX/TX ведомого отключено и SOMI принудительно устанавливается на направление ввода.

Когда STE имеет низкий уровень, функционирование RX/TX ведомого разрешено и SOMI работает как обычно.

#### 41. Режимы ведущего/ведомого SPI.

##### Режим ведущего

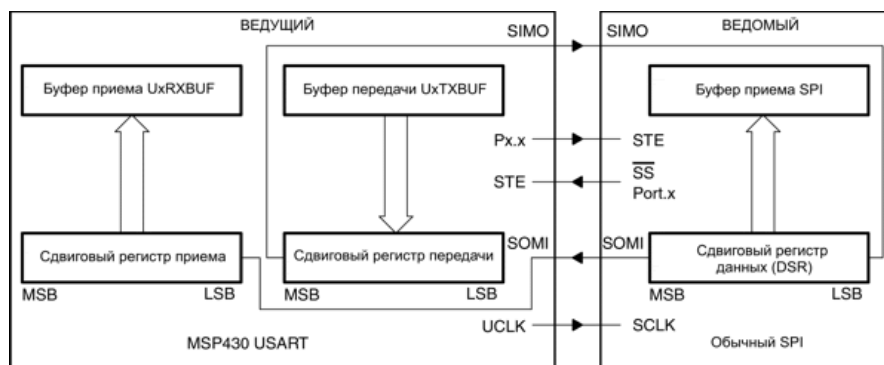


Рис.14-2. USART – ведущий, внешнее устройство - ведомое

На рис.14-2 показан USART в качестве мастера в обеих 3-х и 4-х выводных конфигурациях.

USART инициализирует передачу данных, когда данные перемещаются в буфер передачи данных UxTXBUF. Данные UxTXBUF перемещаются в сдвиговой регистр TX, когда сдвиговой регистр TX пуст, инициируя передачу данных на SIMO, начиная со старшего разряда. Данные на SOMI сдвигаются в сдвиговой регистр приема по противоположному тактовому фронту, начиная со старшего разряда. Когда символ принят, принятые данные перемещены из сдвигового регистра RX в буфер принятых данных UxRXBUF, флаг прерывания приема URXIFGx установлен, указывая завершение операции RX/TX.

Установка флага прерывания передачи UTXIFGx указывает, что данные перемещены из UxTXBUF в сдвиговой регистр TX и UxTXBUF готов для поступления новых данных. Это не указывает на завершение операции RX/TX.

Чтобы принимать данные в USART в режиме ведущего, данные должны быть записаны в UxTXBUF, поскольку операции приема и передачи выполняются одновременно.

##### Режим ведомого

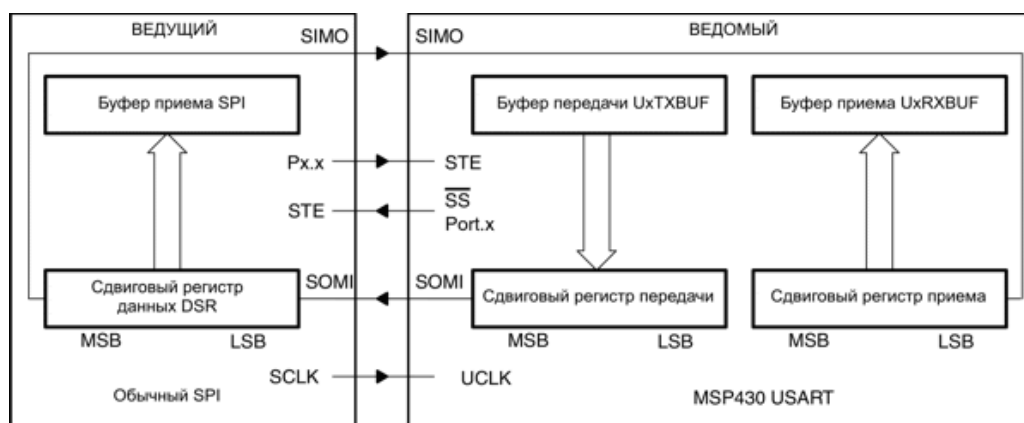


Рис.14-3. USART – ведомый, внешнее устройство - ведущее

На рис.14-3 показан USART в качестве ведомого в обеих 3-х и 4-х выводных конфигурациях.

UCLK используется как вход для тактирования SPI и должен управляться внешним ведущим. Скорость передачи данных определяется этим тактовым сигналом и не зависит от внутреннего генератора скорости передачи. Данные записываются в UxTXBUF и перемещаются в сдвиговый регистр TX до старта передачи UCLK на SOMI. Данные на SIMO сдвигаются в сдвиговый регистр приема по противоположному фронту UCLK и перемещаются в UxRXBUF, когда принято заданное количество бит. Когда данные перемещаются из сдвигового регистра RX в UxRXBUF, устанавливается флаг прерывания URXIFGx, указывая, что данные были приняты. Бит ошибки переполнения OE устанавливается, когда предыдущие принятые данные не были прочитаны из UxRXBUF до перемещения новых данных в UxRXBUF.

## 42. Включение SPI. Разрешение приема/передачи.

Когда USPIEx=0, любая последующая запись в UxTXBUF не приводит к передаче. Данные, записанные в UxTXBUF начнут передаваться, когда USPIEx=1 и активен источник BRCLK. На рис.14-4 и рис.14-5 показаны диаграммы состояний при разрешении передачи.

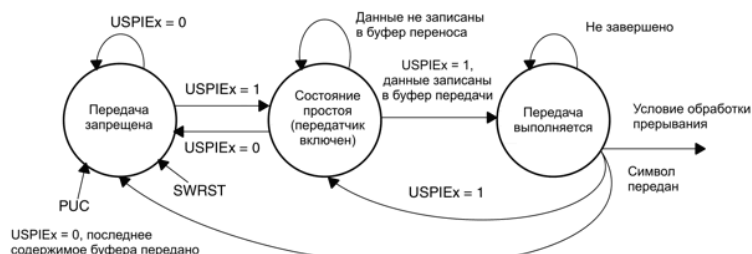


Рис.14-4. Разрешение передачи в режиме ведущего

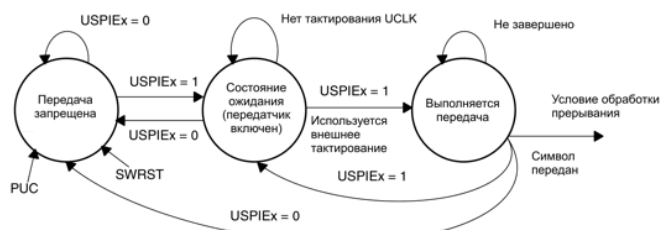


Рис.14-5. Диаграмма состояний разрешения передачи ведомого

## Разрешение приема

Диаграммы состояний разрешения приема SPI показаны на рис.14-6 и рис.14-7. Когда USPIEx=0, UCLK не сдвигает данные в сдвиговый регистр RX.



Рис.14-6. Диаграмма состояний разрешения приема в режиме ведущего SPI

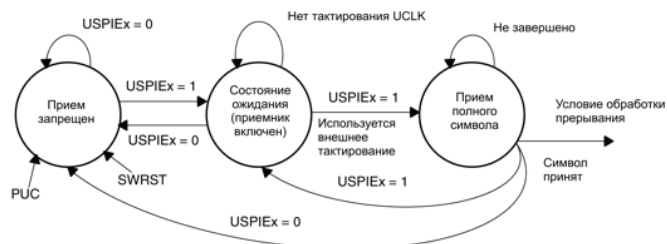


Рис.14-7. Диаграмма состояний разрешения приема ведомым SPI

#### 43. Управление тактированием SPI.

Сигнал UCLK на шине SPI обеспечивается ведущим. Когда MM=1, BITCLK обеспечивается генератором скорости передачи USART на выводе UCLK, как показано на рис.14-8. Когда MM=0, тактирование USART на выводе UCLK обеспечивается ведущим, генератор скорости передачи не используется, а значения битов SSELx не учитываются. Приемник и передатчик SPI работают параллельно и используют одинаковый источник тактирования для передачи данных.

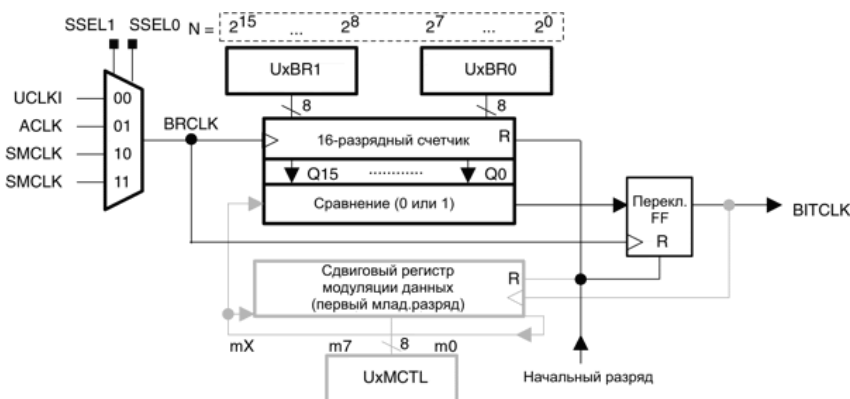


Рис.14-8. Генератор скорости передачи SPI

16-разрядное значение UxBR0+UxBR1 представляет собой коэффициент деления источника тактирования USART – BRCLK. Максимальная скорость передачи, генерируемая в режиме ведущего равна BRCLK/2. Модулятор в генераторе скорости передачи USART не используется в режиме SPI, рекомендуется устанавливать его значение равным 000h. Частота UCLK определяется так:

$$\text{Скорость передачи} = \text{BRCLK} / \text{UxBR}, \text{ где } \text{UxBR} = [\text{UxBR1}, \text{UxBR0}]$$

#### 44. Прерывания SPI.

SPI имеет один вектор прерывания для передачи и один вектор прерывания для приема.

##### Работа прерывания SPI при передаче

Флаг прерывания UTXIFGx устанавливается передатчиком для указания, что UxTXBUF готов к приему другого символа. Запрос прерывания генерируется, если также установлены флаги UTXIEх и GIE. UTXIFGx автоматически сбрасывается, если запрос прерывания обработан или если символ записан в UxTXBUF.

UTXIFGx устанавливается после PUC или когда SWRST=1. UTXIEх сбрасывается после PUC или когда SWRST=1. Это показано на рис.14-10.



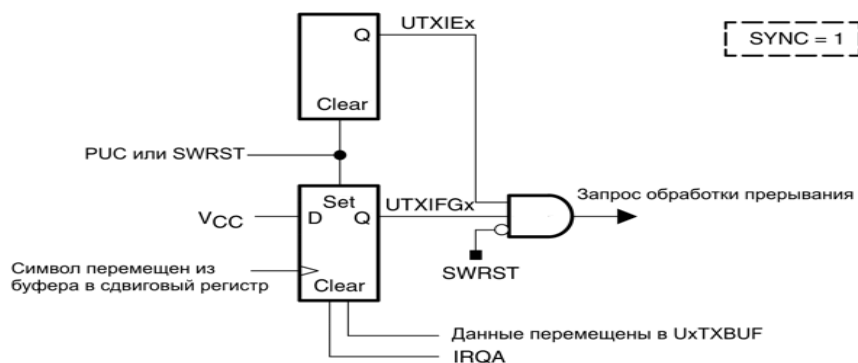


Рис.14-10. Функционирование прерывания при передаче

### Работа прерывания SPI при приеме

Флаг прерывания URXIFGx устанавливается каждый раз, когда символ принят и загружен в UxRXBUF, как показано на рис.14-11 и 14-12. Запрос прерывания генерируется, если также установлены флаги URXIE и GIE. URXIFGx и URXIE сбрасываются сигналом системного сброса PUC или когда SWRST=1. URXIFGx сбрасывается автоматически, если ожидаемое прерывание обработано или когда UxRXBUF прочитан. Это показано на рис.14-11

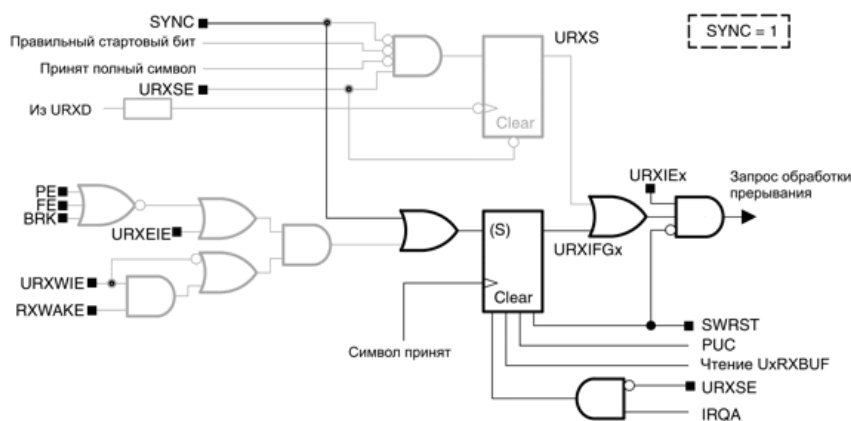


Рис.14-11. Функционирование прерывания при приеме

## 45.Регистры USART, режим SPI.

Регистры USART, показанные в таблице 14-1 и таблице 14-2, структурированы побайтно, поэтому доступ к ним необходимо выполнять с помощью команд работы с байтами.

Таблица 14-1. Регистры управления и статуса USART0

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Регистр управления USART	U0CTL	Чтение/запись	070h	001h после PUC
Регистр управления передачей	U0TCTL	Чтение/запись	071h	001h после PUC
Регистр управления приемом	U0RCTL	Чтение/запись	072h	000h после PUC
Регистр управления модуляцией	U0MCTL	Чтение/запись	073h	Не изменяется
Регистр 0 управления скоростью передачи	U0BR0	Чтение/запись	074h	Не изменяется
Регистр 1 управления скоростью передачи	U0BR1	Чтение/запись	075h	Не изменяется
Регистр буфера приема	U0RXBUF	Чтение	076h	Не изменяется
Регистр буфера передачи	U0TXBUF	Чтение/запись	077h	Не изменяется

Регистр 1 включения модуля SFR*	ME1	Чтение/запись	004h	000h после PUC
Регистр 1 разрешения прерывания SFR*	IE1	Чтение/запись	000h	000h после PUC
Регистр 1 флага прерывания SFR*	IFG1	Чтение/запись	002h	082h после PUC

Таблица 14-2. Регистры управления и статуса USART1

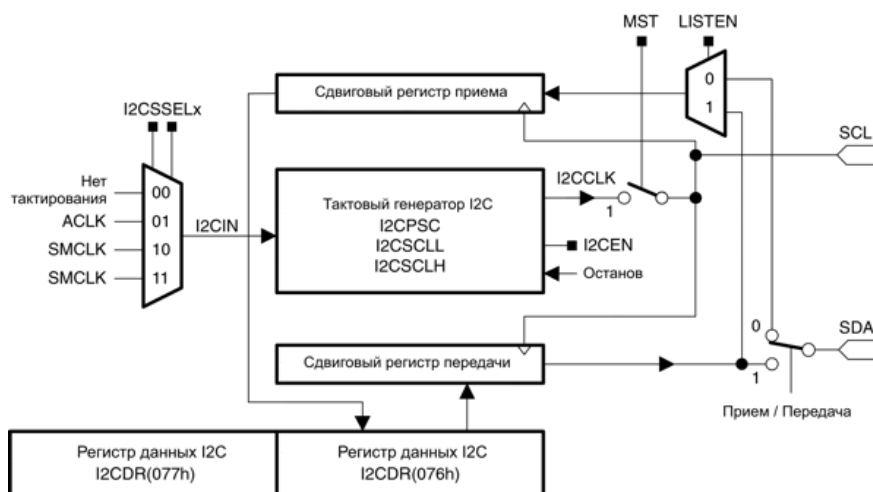
Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Регистр управления USART	U1CTL	Чтение/запись	078h	001h после PUC
Регистр управления передачей	U1TCTL	Чтение/запись	079h	001h после PUC
Регистр управления приемом	U1RCTL	Чтение/запись	07Ah	000h после PUC
Регистр управления модуляцией	U1MCTL	Чтение/запись	07Bh	Не изменяется
Регистр 0 управления скоростью передачи	U1BR0	Чтение/запись	07Ch	Не изменяется
Регистр 1 управления скоростью передачи	U1BR1	Чтение/запись	07Dh	Не изменяется
Регистр буфера приема	U1RXBUF	Чтение	07Eh	Не изменяется
Регистр буфера передачи	U1TXBUF	Чтение/запись	07Fh	Не изменяется
Регистр 2 включения модуля SFR	ME2	Чтение/запись	005h	000h после PUC
Регистр 2 разрешения прерывания SFR	IE2	Чтение/запись	001h	000h после PUC
Регистр 2 флага прерывания SFR	IFG2	Чтение/запись	003h	020h после PUC

Режим SPI имеет следующие возможности:

- 7-ми или 8-разрядные данные
- Работа SPI с 3-мя или 4-мя выводами
- Режимы ведущий или ведомый
- Независимые сдвиговые регистры передачи и приема
- Раздельные буферные регистры передачи и приема
- Выбираемая полярность UCLK и управление фазой
- Программируемая частота UCLK в режиме ведущего
- Независимая возможность прерывания для приема и передачи

#### 46.Периферийный интерфейс USART, режим I2C. Функционирование I2C.

Модуль управления взаимодействием между интегральными схемами (I<sup>2</sup>C) обеспечивает интерфейс между MSP430 и I<sup>2</sup>C-совместимыми устройствами через последовательную двухпроводную шину I<sup>2</sup>C. Внешние компоненты, подключенные к шине I<sup>2</sup>C последовательно передают и принимают последовательные данные в/из USART через 2-х проводной I<sup>2</sup>C-интерфейс.



Модуль I<sup>2</sup>C поддерживает любые ведущие и ведомые устройства, совместимые с I<sup>2</sup>C. На рис.15-2 показан пример шины I<sup>2</sup>C. Каждое устройство обладает уникальным адресом и может работать и как передатчик и как приемник. Устройство, подключенное к шине I<sup>2</sup>C, во время передачи данных может рассматриваться как ведущее или ведомое. Ведущий инициирует передачу данных и генерирует тактовый сигнал SCL. Любое устройство, адресованное ведущим, рассматривается как ведомое.

Рис.15-2. Схема подключений на шине I<sup>2</sup>C

## 47. Инициализация I2C.

Первый, после условия «СТАРТ», байт состоит из 7-разрядного адреса ведомого и бита R/nonW. Когда R/nonW=0, ведущий передает данные ведомому. Когда R/nonW=1, ведущее устройство принимает данные от ведомого. Бит ACK посылается приемником после каждого байта на 9-ом такте SCL.

Рис.15-3. Передача данных модулем I<sup>2</sup>C

#### 48.Режимы адресации I<sup>2</sup>C.

Модуль I<sup>2</sup>C поддерживает 7-разрядный и 10-разрядный режимы адресации.

##### 7-разрядная адресация

В 7-разрядном формате адресации, показанном на рис.15-6, первый байт – это 7-разрядный адрес ведомого и бит R/nonW. Бит ACK посылается приемником после каждого байта.

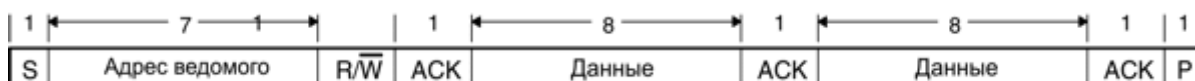


Рис.15-6. 7-разрядный формат модуля I<sup>2</sup>C

##### 10-разрядная адресация

В 10-разрядном адресном формате, показанном на рис.15-7, первый байт содержит 11110b плюс два старших бита 10-разрядного адреса ведомого и бит R/nonW. Бит ACK посылается приемником после каждого байта. Следующий байт содержит оставшиеся 8 бит 10-разрядного адреса ведомого, завершающиеся битом ACK и 8-разрядными данными.

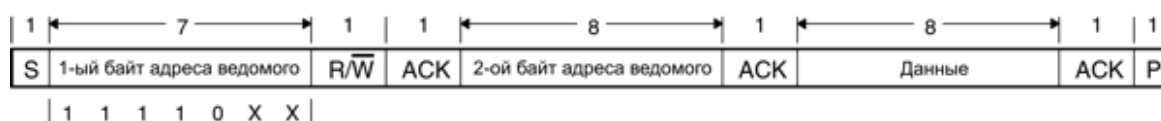


Рис.15-7. 10-разрядный адресный формат модуля I<sup>2</sup>C

#### 49.Режим ведущего I<sup>2</sup>C.

##### Режим ведущего

В режиме ведущего выполнение передачи и приема управляется с помощью битов I<sup>2</sup>CRM, I<sup>2</sup>CSTT и I<sup>2</sup>CSTP, как описано в таблице 15-1. Режим ведущего приемника вводится установкой I<sup>2</sup>CTRX=0 после передачи адресного байта ведомого и установленного бита R/nonW. Режимы ведущего передатчика и ведущего приемника показаны на рис.15-9 и рис.15-10.

SCL удерживается в низком состоянии, когда необходимо вмешательство ЦПУ после передачи байта.

Таблица 15-1. Функционирование ведущего

I2CRM	I2CSTP	I2CSTT	Условие или активность шины
X	0	0	Модуль I <sup>2</sup> C в режиме ведущего, но свободен. Условия «старт» и «стоп» не генерируются.
0	0	1	Активность инициируется установкой I2CSTT. I2CNDAT используется для установки длины передачи. Условие «стоп» автоматически не генерируется после перемещения байт, количество которых задано в I2CNDAT. Программное обеспечение должно установить I2CSTP для генерации условия «стоп» в конце передачи.
0	1	1	I2CNDAT используется для установки длины передачи. Установкой I2CSTT инициируется активность. Условие «стоп» автоматически генерируется после передачи байт, количество которых задано I2CNDAT.
1	0	1	I2CNDAT не используется для установки длины передачи. Длиной передачи должно управлять программное обеспечение. Установка бита I2CSTT инициирует активность. Для инициирования условия «стоп» или останова активности программное обеспечение должно установить бит I2CSTP. Этот режим используется, если необходимо передать более 256 байт.
0	1	0	Установка бита I2CSTP генерирует условие «стоп» на шине после отправки количества байт, заданного I2CNDAT или немедленно, если уже передано количество байт, заданное I2CNDAT.
1	1	0	Установка бита I2CSTP генерирует условие «стоп» на шине после завершения текущей передачи или немедленно, если текущая передача не активна.
1	1	1	Зарезервировано, шина неактивна.

## 50.Режим ведомого I2C.

режиме ведомого операции передачи и приема управляются автоматически модулем I2C.

В режиме ведомого приемника биты последовательных данных принимаются на SDA и сдвигаются по тактовым импульсам, генерируемым ведущим устройством. Ведомое устройство не генерирует тактовый сигнал, но может удерживать линию SCL в состоянии низкого уровня, если после приема байта необходимо вмешательство ЦПУ.

В режим ведомого передатчика можно войти только из режима ведомого приемника. Вход в режим ведомого передатчика происходит, если байт адреса ведомого, переданный ведущим, является таким же адресом, как и его собственный и был послан установленный бит R/W, указывая на запрос отправки данных ведущему. Ведомый передатчик сдвигает последовательные данные из устройства на SDA по импульсам тактирования, генерируемым ведущим устройством. Ведомое устройство не генерирует тактовых сигналов, но может удерживать линию SCL в состоянии низкого уровня, если после передачи байта необходимо вмешательство ЦПУ.

## 51.Прерывания I2C.

Модуль I<sup>2</sup>C имеет один вектор прерывания для восьми флагов прерывания. Каждый флаг прерывания имеет собственный бит разрешения прерывания. Когда прерывание разрешено и установлен бит GIE, флаг прерывания будет генерировать запрос прерывания. Следующие события вызывают I<sup>2</sup>C прерывание:

Флаг	Условие прерывания
------	--------------------

прерывания	
ALIFG	Потеря арбитража. Арбитраж может быть потерян, когда два или более передатчиков начинают передачу одновременно или когда программное обеспечение пытается инициировать I <sup>2</sup> C передачу при I2CBB=1. Флаг ALIFG устанавливается, когда арбитраж потерян. Когда ALIFG установлен, биты MST и I2CSTP очищаются и контроллер I <sup>2</sup> C становится ведомым приемником.
NACKIFG	Прерывание при отсутствии подтверждения. Этот флаг устанавливается, когда ведущий не получил от ведомого подтверждение. NACKIFG используется только в режиме ведущего.
OAIFG	Прерывание собственного адреса. Флаг прерывания OAIFG устанавливается, когда другой ведущий адресует I <sup>2</sup> C модуль. OAIFG используется только в режиме ведомого.
ARDYIFG	Прерывание «регистр доступен». Этот флаг устанавливается, когда ранее запрограммированный перенос завершен, а биты статуса обновлены. Это прерывание используется для уведомления ЦПУ о том, что регистры I <sup>2</sup> C готовы к доступу.
RXRDIYIFG	Прерывание/статус готовности приема. Этот флаг устанавливается, когда модуль I <sup>2</sup> C принял новые данные. RXRDIYIFG автоматически очищается, когда I2CDR прочитан и буфер приема пуст. Переполнение приемника показывается, если бит I2CRXOVR=1. RXRDIYIFG используется только в режиме приема.
TXRDIYIFG	Прерывание/статус готовности передачи. Регистр I2CDR готов к новой передаче данных, когда I2CNDAT>0 или I2CRM=1 (режим передачи ведущего) или же когда другой ведущий запрашивает данные (режим передачи ведомого). TXRDIYIFG автоматически очищается, когда I2CDR и буфер передачи полны. Опустошение передачи показывается, если I2CTXUDF=1. Не используется в режиме приема.
GCIFG	Прерывание общего вызова. Этот флаг устанавливается, когда модуль I <sup>2</sup> C принял адрес общего вызова (00h). GCIFG используется только в режим приема.
STTIFG	Прерывание при обнаружении условия старта. Этот флаг устанавливается, когда модуль I <sup>2</sup> C обнаружил условие старта в режиме ведомого. Это позволяет MSP430 находиться в режиме пониженного энергопотребления с неактивным источником тактирования I <sup>2</sup> C до инициирования связи ведущим по I <sup>2</sup> C. STTIFG

## 52.Регистры I<sup>2</sup>C.

Регистры модуля I<sup>2</sup>C и соответствующие адреса приведены в таблице 15-3.

Таблица 15-3. Регистры I<sup>2</sup>C

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Разрешение прерывания I <sup>2</sup> C	I2CIE	Чтение/запись	050h	Сброс с PUC
Флаг прерывания I <sup>2</sup> C	I2CIFG	Чтение/запись	051h	Сброс с PUC
Подсчет данных I <sup>2</sup> C	I2CNDAT	Чтение/запись	052h	Сброс с PUC
Управление USART	U0CTL	Чтение/запись	070h	Сброс с PUC
Управление передачей I <sup>2</sup> C	I2CTCTL	Чтение/запись	071h	Сброс с PUC
Управление данными I <sup>2</sup> C	I2CDCTL	Только чтение	072h	Сброс с PUC
Прескалер I <sup>2</sup> C	I2CPSC	Чтение/запись	073h	Сброс с PUC
«Высокий» SCL I <sup>2</sup> C	I2CSCLH	Чтение/запись	074h	Сброс с PUC
«Низкий» SCL I <sup>2</sup> C	I2CSCLL	Чтение/запись	075h	Сброс с PUC
Данные I <sup>2</sup> C	I2CDR	Чтение/запись	076h	Сброс с PUC
Собственный адрес I <sup>2</sup> C	I2COA	Чтение/запись	0118h	Сброс с PUC
Адрес ведомого I <sup>2</sup> C	I2CSA	Чтение/запись	011Ah	Сброс с PUC
Вектор прерываний I <sup>2</sup> C	I2CIV	Только чтение	011Ch	Сброс с PUC

## 53.Компаратор А. Структура и функционирование компаратора.

Компаратор А – это аналоговый компаратор напряжения. В этом разделе описывается компаратор А. Компаратор А реализован в устройствах MSP430x11x1, MSP430x12x, MSP430x13x, MSP430x14x, MSP430x15x и MSP430x16x.

Модуль компаратора А поддерживает высокоточные аналого-цифровые преобразования напряжения, контроль напряжения питания и мониторинг внешних аналоговых сигналов. Блок-схема компаратора А показана на рис.16-1.

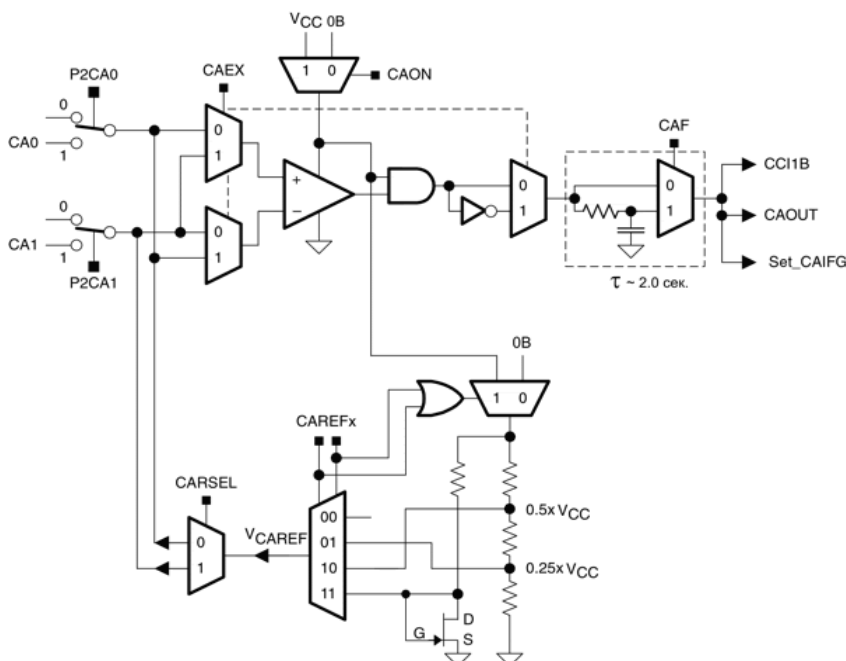


Рис.16-1 Блок-схема компаратора А

Компаратор сравнивает аналоговые напряжения на входах «+» и «-». Если вход «+» более положителен, чем вход «-», на выходе компаратора CAOUT появляется сигнал высокого уровня. Компаратор может быть включен или выключен с помощью управляющего бита CAON. Если компаратор не используется, для уменьшения потребляемого тока его необходимо выключать. Когда компаратор выключен, на выходе CAOUT всегда сигнал низкого уровня.

#### 54.Регистры компаратора.

Регистры компаратора А приведены в таблице 16-1.

Таблица 16-1. Регистры компаратора А

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Управляющий регистр 1 компаратора А	CACTL1	Чтение/запись	059h	Сброс с POR
Управляющий регистр 2 компаратора А	CACTL2	Чтение/запись	05Ah	Сброс с POR
Отключение порта компаратора А	CAPD	Чтение/запись	05Bh	Сброс с POR

#### CACTL1, регистр управления 1 компаратора А

76543210

CAEX	CARSEL	CAREF <sub>x</sub>	CAON	CAIES	CAIE	CAIFG
------	--------	--------------------	------	-------	------	-------

rw-(0) rw-(0) rw-(0) rw-(0) rw-(0) rw-(0) rw-(0) rw-(0)

### 55. Модуль АЦП12. Структура и функционирование.

Модуль АЦП12 представляет собой высокоэффективный 12-разрядный аналого-цифровой преобразователь. В этом разделе описывается АЦП12. АЦП12 реализован в устройствах MSP430x13x, MSP430x14x, MSP430x15x и MSP430x16x.

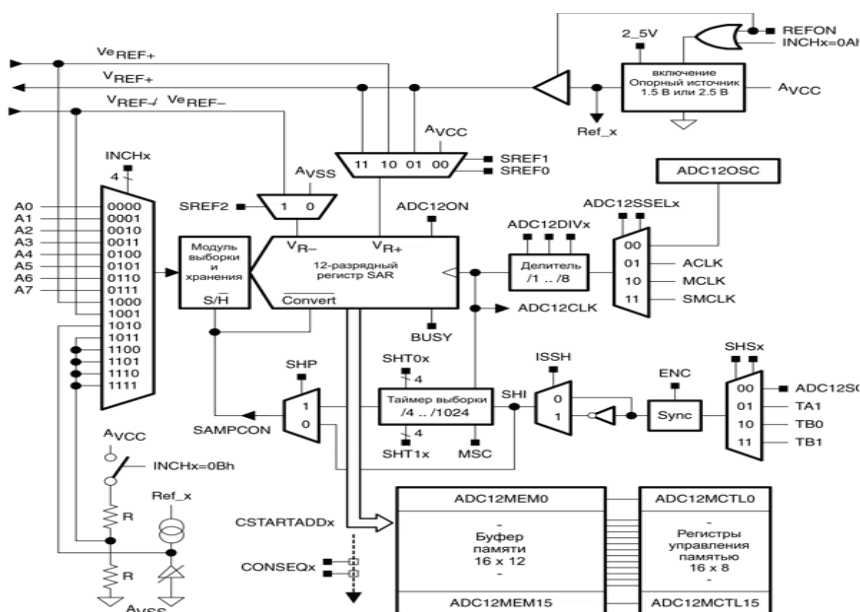


Рис.17-1 Блок-схема АЦП12

Ядро АЦП преобразует аналоговый входной сигнал в 12-разрядное цифровое представление и сохраняет результат в памяти преобразований. Ядро использует два программируемых/выбираемых уровня напряжения (VR+ и VR-) для задания верхнего и нижнего пределов преобразования. На цифровом выходе (NADC) представлена полная шкала (0FFFh), когда входной сигнал равен или выше VR+, и ноль, когда входной сигнал равен или ниже VR-. Входной канал и опорные уровни напряжения (VR+ и VR-) задаются в памяти управления преобразованиями. Формула преобразования для результата АЦП NADC выглядит следующим образом:



$$N_{ADC} = 4095 \times \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

## 56. Выбор тактирования преобразования, выбор аналогового порта, генератор опорного напряжения.

### Выбор тактирования преобразования

ADC12CLK используется как для тактирования преобразования, так и для генерации периода выборки, когда выбран импульсный режим выборки. Для выбора источника тактирования ADC12 используются биты ADC12SSELx, а частота выбранного источника может быть поделена на 1-8 с помощью битов ADC12DIVx. Возможно использование следующих источников ADC12CLK: SMCLK, MCLK, ACLK и внутреннего осциллятора ADC12OSC.

ADC12OSC, генерируемый внутренне, лежит в диапазоне 5 МГц, но варьируется в зависимости от конкретного устройства, напряжения питания и температуры. См. справочное руководство конкретного устройства для уточнения значения ADC12OSC.

Пользователь должен гарантировать, что выбранный источник тактирования для ADC12CLK останется активным до конца преобразования. Если тактовые сигналы будут сняты во время преобразования, операция не будет завершена и любой результат будет неверным.

### Выбор аналогового порта

Входы АЦП12 мультиплексированы с ножками порта P6, имеющими цифровые КМОП ячейки. Когда аналоговые сигналы прикладываются к цифровым КМОП-схемам, может течь паразитный ток от VCC к GND. Этот паразитный ток появляется, если величина входного напряжения находится около переходного уровня ячейки. Отключение буфера ножки порта устраняет протекание паразитного тока и вследствие этого уменьшает общий потребляемый ток. Биты P6SELx дают возможность отключать входные буферы ножки порта.

### Генератор опорного напряжения

Модуль АЦП12 содержит встроенный генератор опорного напряжения с двумя выбираемыми уровнями напряжения: 1,5 В и 2,5 В. Любое из этих опорных напряжений может быть использовано внутренне или внешне на выводе VREF+.

Установкой REFON=1 включается внутренний опорный источник. Когда REF2\_5V=1, внутреннее опорное напряжение равно 2,5 В, при REF2\_5V=0 опорное напряжение равно 1,5 В. Если генератор опорного напряжения не используется, он может быть выключен для уменьшения потребления энергии.

Для правильной работы внутреннего генератора опорного напряжения необходимо использовать емкость временного хранения энергии, подключенную между VREF+ и AVSS. Рекомендуется в качестве такой емкости использовать комбинацию из включенных параллельно конденсаторов на 10 мкФ и 0,1 мкФ. После включения в течение максимум 17 мС необходимо дать возможность генератору опорного напряжения зарядить конденсаторы хранения энергии. Если внутренний опорный генератор не используется при преобразованиях, конденсаторы не требуются.

## 57. Расширенный и импульсный режим выборки.

### Расширенный режим выборки

Расширенный режим выборки выбирается, когда  $SHP=0$ . Сигнал  $SHI$  напрямую управляет  $SAMPCON$  и определяет длительность периода выборки  $t_{sample}$ . Когда  $SAMPCON$  имеет высокий уровень, выборка активна. Переход сигнала  $SAMPCON$  с высокого уровня на низкий запускает преобразование после синхронизации с  $ADC12CLK$ . См. рис.17-3.

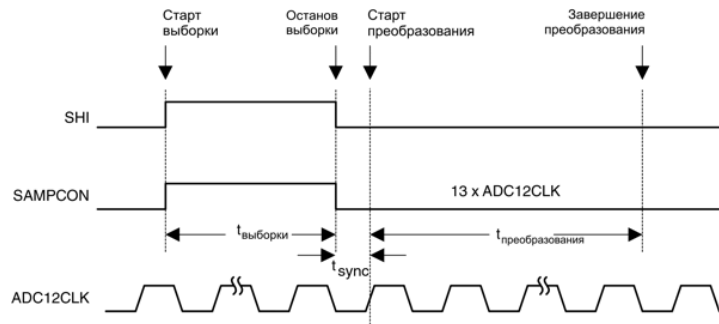


Рис.17-3 Расширенный режим выборки

### Импульсный режим выборки

Импульсный режим выборки выбирается, когда  $SHP=0$ . Сигнал  $SHI$  используется для запуска таймера выборки. Биты  $SHT0x$  и  $SHT1x$  в  $ADC12CTL0$  управляют интервалом таймера выборки, который задает период  $t_{sample}$  выборки  $SAMPCON$ . Таймер выборки оставляет высокий уровень  $SAMPCON$  после синхронизации с  $ADC12CLK$  для запрограммированного интервала  $t_{sample}$ . Общее время выборки равно  $t_{sample}$  плюс  $t_{sync}$ . См. рис.17-4.

Биты  $SHTx$  устанавливают время выборки в 4 раза больше чем  $ADC12CLK$ .  $SHT0x$  устанавливает время выборки для  $ADC12MCTL0-7$ , а  $SHT1x$  устанавливает время выборки для  $ADC12MCTL8-15$ .

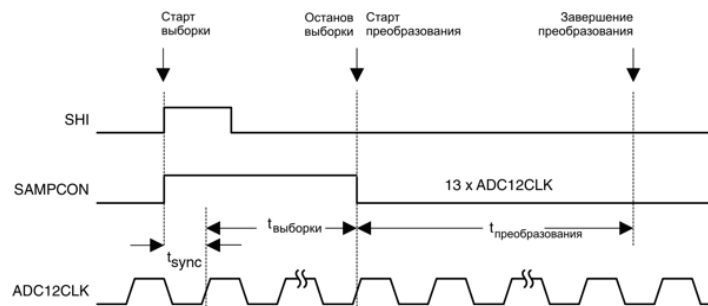


Рис.17-4 Импульсный режим выборки

## 58. Память преобразований. Режимы преобразований АЦП12.

Результаты преобразований сохраняются в 16-ти регистрах памяти преобразований  $ADC12MEMx$ . Каждый регистр  $ADC12MEMx$  конфигурируется соответствующим управляющим регистром  $ADC12MCTLx$ . Биты  $SREFx$  устанавливают опорное напряжение, а биты  $INCHx$  задают входной канал. Бит  $EOS$  определяет конец последовательности, когда используется последовательный режим преобразования. Следующие друг за другом преобразования последовательно сохраняются в регистрах с  $ADC12MEM15$  по  $ADC12MEM0$ , когда бит  $EOS$  в  $ADC12MCTL15$  не установлен.

Биты  $CSTARTADDx$  определяют первый регистр  $ADC12MCTLx$ , используемый для любого преобразования. Если выбраны одноканальный или повторный одноканальный режимы преобразования,  $CSTARTADDx$  указывают на единственный  $ADC12MCTLx$ , который будет использован.

Если выбран режим преобразования «последовательность каналов» или «повторяющаяся

последовательность каналов», CSTARTADDx указывают на расположение ADC12MCTLx, который будет использоваться в последовательности. Программно невидимый указатель автоматически инкрементируется до следующего ADC12MCTLx в последовательности после каждого завершения преобразования. Последовательность продолжается до обработки бита EOS в ADC12MCTLx – это будет обработка последнего управляющего байта.

Когда результат преобразования записывается в выбранный регистр ADC12MEMx, устанавливается соответствующий флаг в регистре ADC12IFGx.

### Режимы преобразований АЦП12

АЦП12 имеет четыре режима работы, выбираемые битами CONSEQx так, как описано в таблице 17-1.

Таблица 17-1. Сводный перечень режимов преобразования

CONSEQx	Режим	Операция
00	Одноканальный с одиночным преобразованием	Выполняется одно преобразование в одном канале.
01	Последовательность каналов	Выполняются однократные преобразования последовательности каналов.
10	Повторяющийся одноканальный	Выполняется повторяющееся преобразование в одном канале.
11	Повторяющаяся последовательность каналов	Выполняются повторяющиеся преобразования последовательности каналов.

#### 59. Одноканальный режим с одиночным преобразованием.

В одном канале однократно выполняется выборка и преобразование. Результат АЦП записывается в регистр ADC12MEMx, определенный битами CSTARTADDx. На рис.17-6 показан процесс одноканального режима с одиночным преобразованием. Если преобразования запускаются ADC12SC, поочередные преобразования могут быть запущены битом ADC12SC. Когда используется другой источник запуска, ENC должен переключаться между каждым преобразованием.

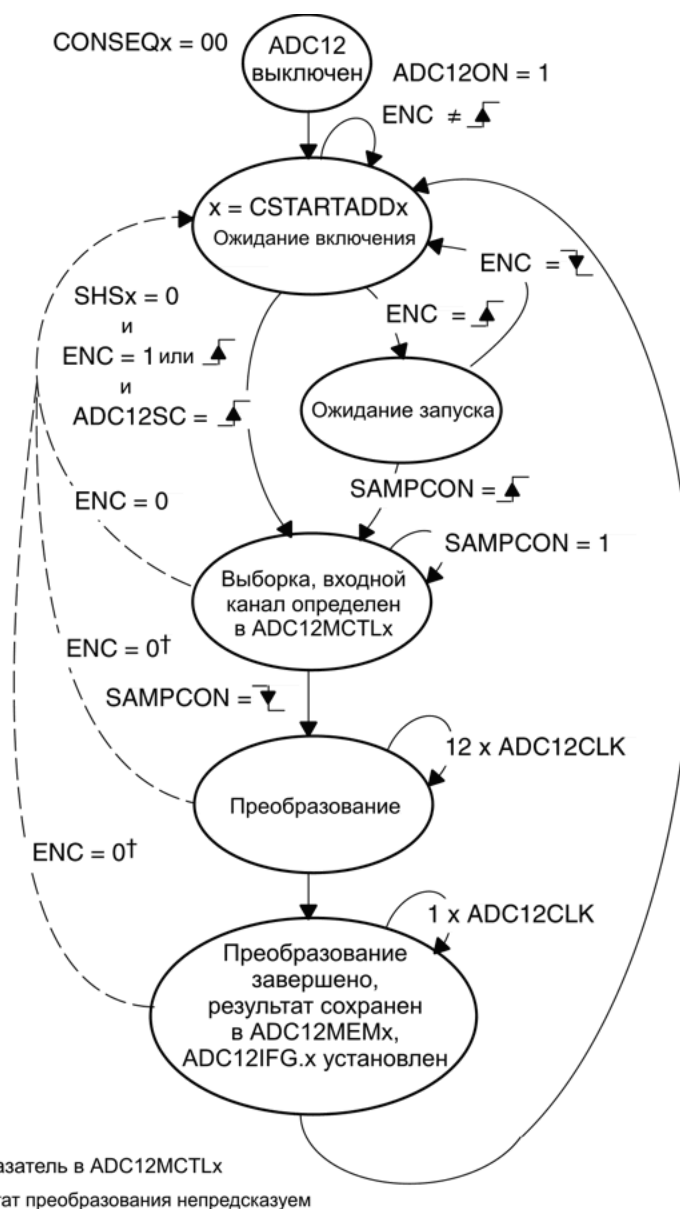


Рис.17-6 Одноканальный режим одиночного преобразования

## 60. Режим последовательности каналов.

В режиме последовательности каналов однократно выполняется выборка и преобразование. Результат АЦП записывается в память преобразований, начиная с ADCMEMx, определенным битами CSTARTADDx. Последовательность останавливается после измерения в канале с установленным битом EOS. На рис.17-7 показан режим последовательности каналов. Если последовательность запускает ADC12SC, поочередные последовательности могут запускаться битом ADC12SC. Когда используется другой источник запуска, ENC должен переключаться между каждой последовательностью.

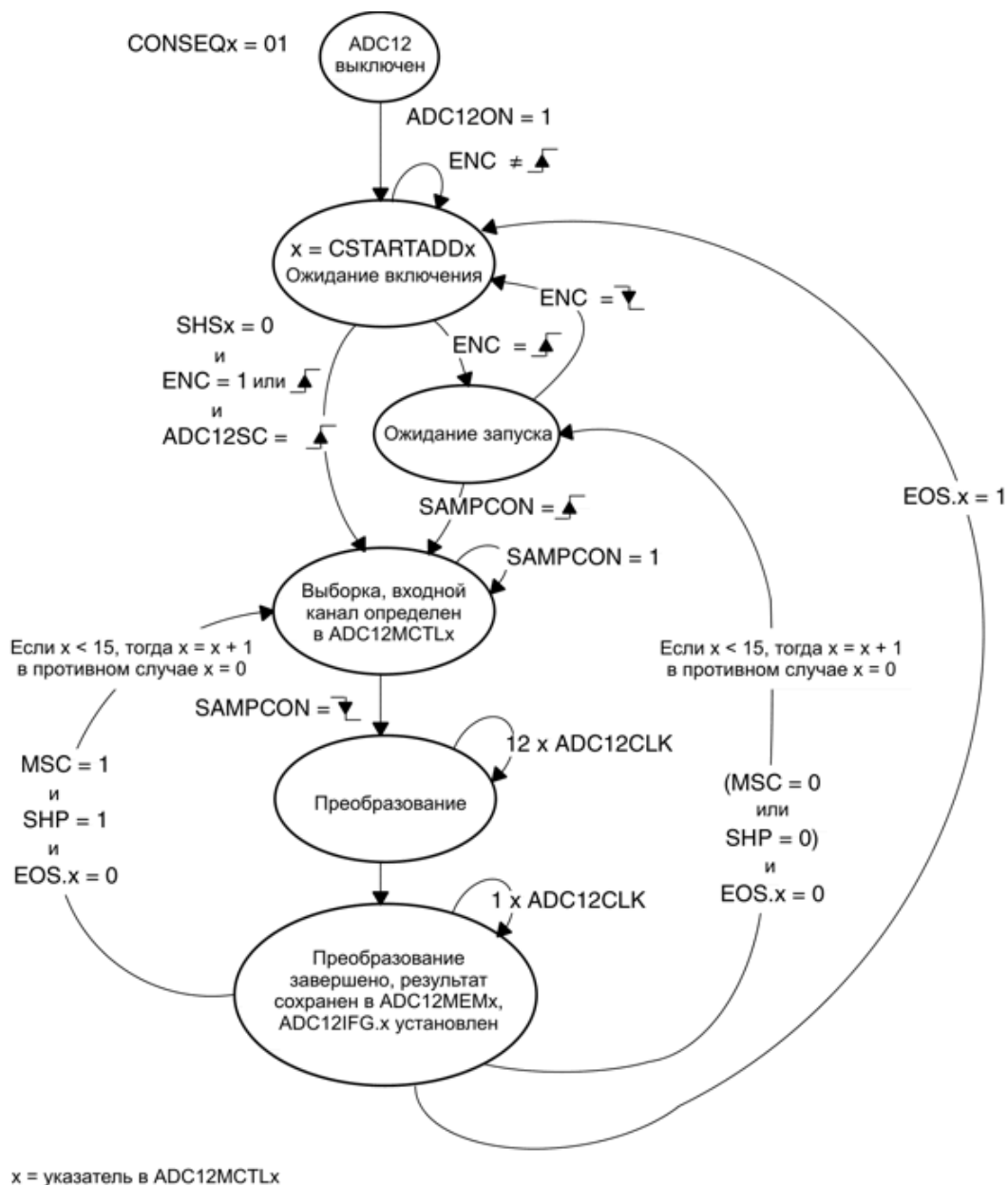


Рис.17-7 Режим последовательности каналов

### 61.Повторяющийся одноканальный режим.

В одном канале непрерывно выполняются выборка и преобразование. Результат АЦП записывается в ADC12MEMx, определенный битами CSTARTADDx. Необходимо считывать результат после завершения преобразования, потому что используется только один регистр памяти ADC12MEMx, перезаписываемый с каждым новым преобразованием. На рис.17-8 показан повторяющийся одноканальный режим.

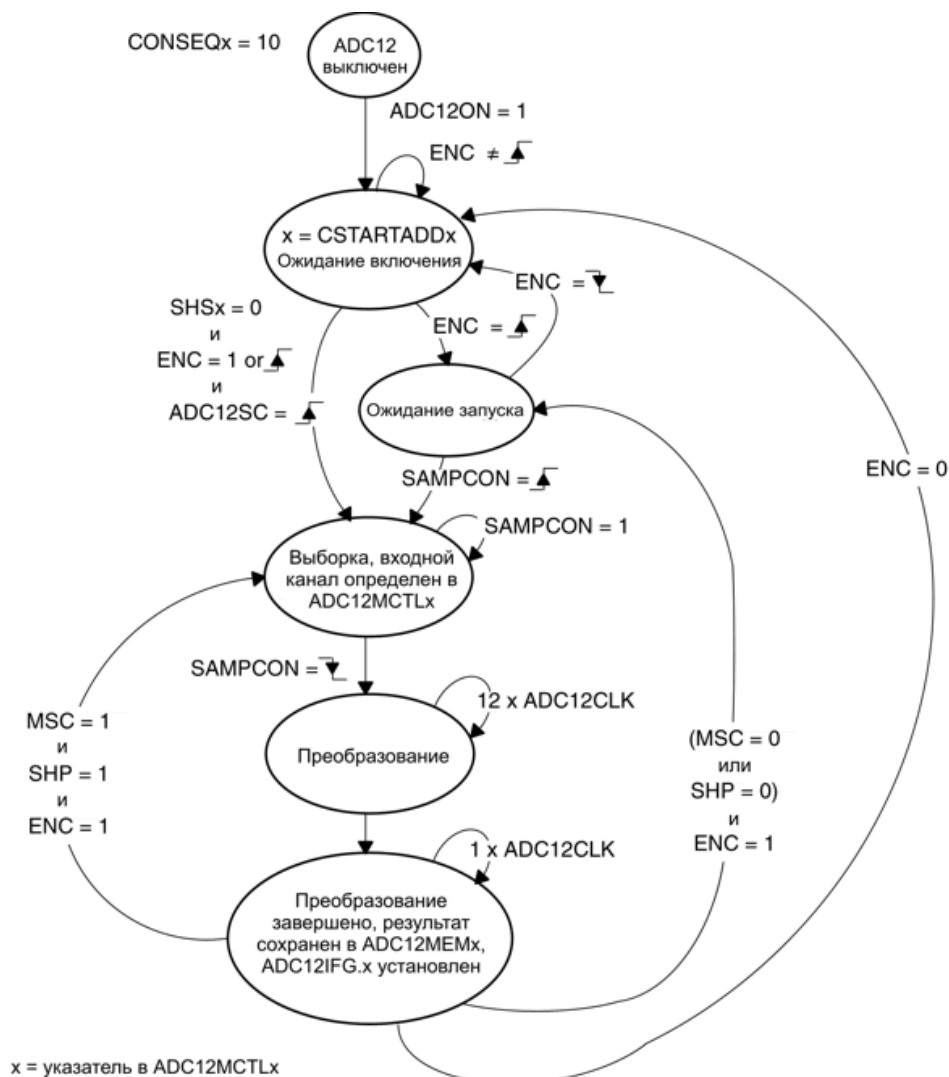


Рис.17-8 Повторяющийся одноканальный режим

## 62.Режим повторяющейся последовательности каналов.

Непрерывно выполняются выборка и преобразование последовательности каналов. Результат АЦП записывается в память преобразований, начиная с ADC12MEMx, определенного битами CSTARTADDx. Последовательность останавливается после измерения в канале с установленным битом EOS и стартует снова по следующему сигналу запуска. На рис.17-9 показан режим повторяющейся последовательности каналов.

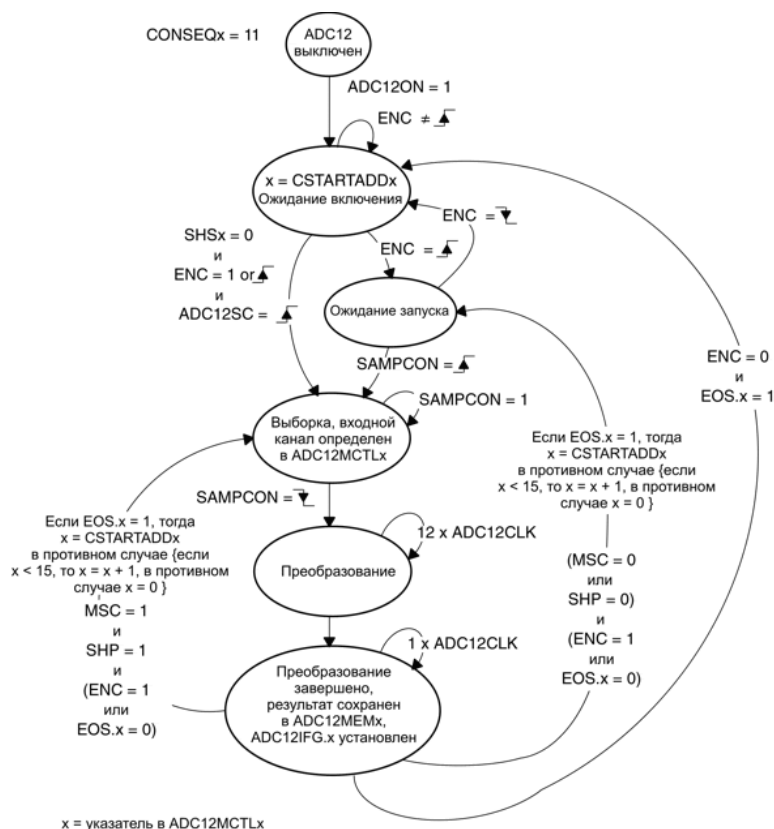


Рис.17-9 Режим повторяющейся последовательности каналов

### 63.Прерывания АЦП12.

АЦП12 имеет 18 источников прерывания:

- ADC12IFG0-ADC12IFG15
- ADC12OV, переполнение ADC12MEMx
- ADC12TOV, переполнение времени преобразования АЦП12

Биты ADC12IFGx устанавливаются, когда в их соответствующие регистры памяти ADC12MEMx загружается результат преобразования. Если соответствующий бит ADC12IEx и бит GIE установлены, генерируется запрос прерывания. Состояние ADC12OV появляется, когда результат преобразования записывается в любой регистр ADC12MEMx до прочтения предыдущего результата. Состояние ADC12TOV генерируется, когда до завершения текущего преобразования затребована другая выборка-преобразование.

### 64.Регистры АЦП12.

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Управляющий регистр 0 АЦП12	ADC12CTL0	Чтение/запись	01A0h	Сброс с POR
Управляющий регистр 1 АЦП12	ADC12CTL1	Чтение/запись	01A2h	Сброс с POR
Регистр флагов прерываний АЦП12	ADC12IFG	Чтение/запись	01A4h	Сброс с POR
Регистр разрешения прерываний АЦП12	ADC12IE	Чтение/запись	01A6h	Сброс с POR
Слово вектора прерываний АЦП12	ADC12IV	Чтение	01A8h	Сброс с POR
Регистр памяти 0 АЦП12	ADC12MEM0	Чтение/запись	0140h	Не изменяется
Регистр памяти 1 АЦП12	ADC12MEM1	Чтение/запись	0142h	Не изменяется

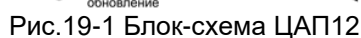
Регистр памяти 2 АЦП12	ADC12MEM2	Чтение/запись	0144h	Не изменяется
Регистр памяти 3 АЦП12	ADC12MEM3	Чтение/запись	0146h	Не изменяется
Регистр памяти 4 АЦП12	ADC12MEM4	Чтение/запись	0148h	Не изменяется
Регистр памяти 5 АЦП12	ADC12MEM5	Чтение/запись	014Ah	Не изменяется
Регистр памяти 6 АЦП12	ADC12MEM6	Чтение/запись	014Ch	Не изменяется
Регистр памяти 7 АЦП12	ADC12MEM7	Чтение/запись	014Eh	Не изменяется
Регистр памяти 8 АЦП12	ADC12MEM8	Чтение/запись	0150h	Не изменяется
Регистр памяти 9 АЦП12	ADC12MEM9	Чтение/запись	0152h	Не изменяется
Регистр памяти 10 АЦП12	ADC12MEM10	Чтение/запись	0154h	Не изменяется
Регистр памяти 11 АЦП12	ADC12MEM11	Чтение/запись	0156h	Не изменяется
Регистр памяти 12 АЦП12	ADC12MEM12	Чтение/запись	0158h	Не изменяется
Регистр памяти 13 АЦП12	ADC12MEM13	Чтение/запись	015Ah	Не изменяется
Регистр памяти 14 АЦП12	ADC12MEM14	Чтение/запись	015Ch	Не изменяется
Регистр памяти 15 АЦП12	ADC12MEM15	Чтение/запись	015Eh	Не изменяется
Управление регистром памяти 0 АЦП12	ADC12MCTL0	Чтение/запись	080h	Сброс с POR
Управление регистром памяти 1 АЦП12	ADC12MCTL1	Чтение/запись	081h	Сброс с POR
Управление регистром памяти 2 АЦП12	ADC12MCTL2	Чтение/запись	082h	Сброс с POR
Управление регистром памяти 3 АЦП12	ADC12MCTL3	Чтение/запись	083h	Сброс с POR
Управление регистром памяти 4 АЦП12	ADC12MCTL4	Чтение/запись	084h	Сброс с POR
Управление регистром памяти 5 АЦП12	ADC12MCTL5	Чтение/запись	085h	Сброс с POR
Управление регистром памяти 6 АЦП12	ADC12MCTL6	Чтение/запись	086h	Сброс с POR
Управление регистром памяти 7 АЦП12	ADC12MCTL7	Чтение/запись	087h	Сброс с POR
Управление регистром памяти 8 АЦП12	ADC12MCTL8	Чтение/запись	088h	Сброс с POR
Управление регистром памяти 9 АЦП12	ADC12MCTL9	Чтение/запись	089h	Сброс с POR
Управление регистром памяти 10 АЦП12	ADC12MCTL10	Чтение/запись	08Ah	Сброс с POR
Управление регистром памяти 11 АЦП12	ADC12MCTL11	Чтение/запись	08Bh	Сброс с POR
Управление регистром памяти 12 АЦП12	ADC12MCTL12	Чтение/запись	08Ch	Сброс с POR
Управление регистром памяти 13 АЦП12	ADC12MCTL13	Чтение/запись	08Dh	Сброс с POR
Управление регистром памяти 14 АЦП12	ADC12MCTL14	Чтение/запись	08Eh	Сброс с POR
Управление регистром памяти 15 АЦП12	ADC12MCTL15	Чтение/запись	08Fh	Сброс с POR

#### 65.ЦАП12. Структура и функционирование. (В ПАЧКЕ С СИСТЕМНЫМ АНАЛИЗОМ!!!)

Модуль АЦП12 представляет собой 12-разрядный ЦАП на основе матрицы резисторов R-2R. ЦАП12 может быть сконфигурирован в 8-ми или 12-разрядном режиме и может использоваться совместно с контроллером DMA. Когда в устройстве представлено несколько модулей ЦАП12, они могут быть сгруппированы вместе для синхронного обновления.

АП12 может быть сконфигурирован на работу в 8-ми или 12-разрядном режиме с помощью бита DASC12RES. Кроме того, полный диапазон вывода программируется через бит DAC12IR и может быть 1х или 3х-кратен выбранному опорному напряжению. Эта возможность позволяет пользователю управлять динамическим диапазоном ЦАП12. Когда используется внутренний опорный источник, полный диапазон вывода всегда равен 1х опорного напряжения. Бит DAC12DF позволяет пользователю выбирать для ЦАП натуральные двоичные данные или данные с дополнением до двух.





### 23.2.1. Ядро 12-битного ЦАП

Таблица 23.1. Выходной диапазон ЦАП ( $V_{\text{ref}} = V_{e\text{REF}+}$  или  $V_{\text{REF}+}$ )

Разрешение	DAC12RES	DAC12IR	Выходное напряжение
12 бит	0	0	$V_{\text{out}} = V_{\text{ref}} \times 3 \times \frac{\text{DAC12\_xDAT}}{4096}$
12 бит	0	1	$V_{\text{out}} = V_{\text{ref}} \times \frac{\text{DAC12\_xDAT}}{4096}$
8 бит	1	0	$V_{\text{out}} = V_{\text{ref}} \times 3 \times \frac{\text{DAC12\_xDAT}}{256}$
8 бит	1	1	$V_{\text{out}} = V_{\text{ref}} \times \frac{\text{DAC12\_xDAT}}{256}$

В 8-битном режиме наибольшее используемое значение регистра DAC12\_xDAT составляет 0FFh, а в 12-битном режиме — 0FFFh. Значения, превышающие указанные величины, могут быть записаны в регистр, однако все неиспользуемые старшие биты будут проигнорированы.

## Выбор порта ЦАП12

Выходы ЦАП12 мультиплексированы с ножками порта P6 и аналоговыми входами АЦП12. Когда  $DAC12AMPx > 0$ , для ножек автоматически выбирается функция ЦАП12, независимо от состояния связанных с ними битов P6SELx и P6DIRx.

### 19.2.2 Опорный источник ЦАП12

Опорный источник для ЦАП12 конфигурируется для использования либо двух внешних опорных напряжений, либо внутреннего опорного источника 1.5В/2.5В от модуля АЦП12 с помощью битов DAC12SREFx. Когда  $DAC12SREFx = \{0,1\}$ , как опорный используется сигнал VREF+, а когда  $DAC12SREFx = \{2,3\}$ , в качестве опорного используется сигнал VeREF+.

При использовании внутреннего опорного источника АЦП12, он должен быть включен и сконфигурирован через соответствующие управляющие биты АЦП12 (см. раздел «АЦП12»). Как только опорный источник АЦП12 сконфигурирован, опорное напряжение подается на VREF+.

#### 67. Калибровка смещения выходного усилителя ЦАП12.

Напряжение смещения выходного усилителя ЦАП12 может быть положительным или отрицательным. Когда смещение отрицательное, выходной усилитель пытается управлять отрицательным напряжением, но не может этого сделать. Выходное напряжение остается равным нулю, пока цифровой вход ЦАП12 не обеспечит достаточного для преодоления отрицательного напряжения смещения положительного выходного напряжения. Получающаяся передаточная функция показана на рис.19-4.



Рис.19-4 Отрицательное смещение

Когда выходной усилитель имеет положительное смещение, ноль на цифровом входе не позволяет получить нулевое выходное напряжение. Выходное напряжение ЦАП12 достигает максимального выходного сигнала до того момента, когда данные на входе ЦАП12 достигнут максимального кода. Это показано на рис.19-5.

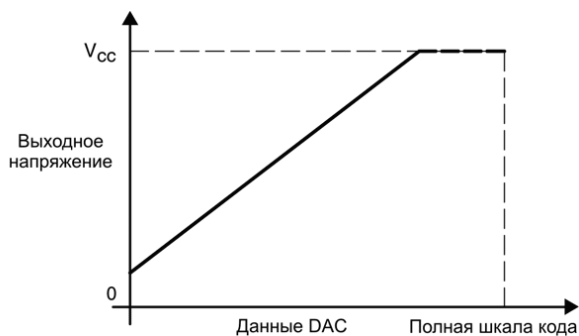


Рис.19-5 Положительное смещение

ЦАП12 имеет возможность калибровки напряжения смещения выходного усилителя. Установка бита DAC12CALON инициирует калибровку смещения. Калибровка должна быть завершена до

использования ЦАП12. Когда калибровка выполнена, бит DAC12CALON автоматически сбрасывается. Биты DAC12AMPx должны быть сконфигурированы до калибровки.

## 68.Прерывания и регистры ЦАП12

Вектор прерываний ЦАП12 является общим с контроллером DMA. Программное обеспечение должно проверять флаги DAC12IFG и DMAIFG для определения источника прерывания.

Бит DAC12IFG устанавливается, когда DAC12xLSELx>0 и данные ЦАП12 защелкнуты от регистра DAC12\_xDAT в защелке данных. Когда DAC12xLSELx=0, флаг DAC12IFG не устанавливается.

Установленный бит DAC12IFG показывает, что ЦАП12 готов для приема новых данных. Если установлены оба бита DAC12IE и GIE, DAC12IFG генерирует запрос прерывания. Флаг DAC12IFG не сбрасывается автоматически. Его должно сбрасывать программное обеспечение.

Регистры:

Регистр	Краткое обозначение	Тип регистра	Адрес	Исходное состояние
Управление DAC12_0	DAC12_0CTL	Чтение/запись	01C0h	Сброс с POR
Данные DAC12_0	DAC12_0DAT	Чтение/запись	01C8h	Сброс с POR
Управление DAC12_1	DAC12_1CTL	Чтение/запись	01C2h	Сброс с POR
Данные DAC12_1	DAC12_1DAT	Чтение/запись	01CAh	Сброс с POR