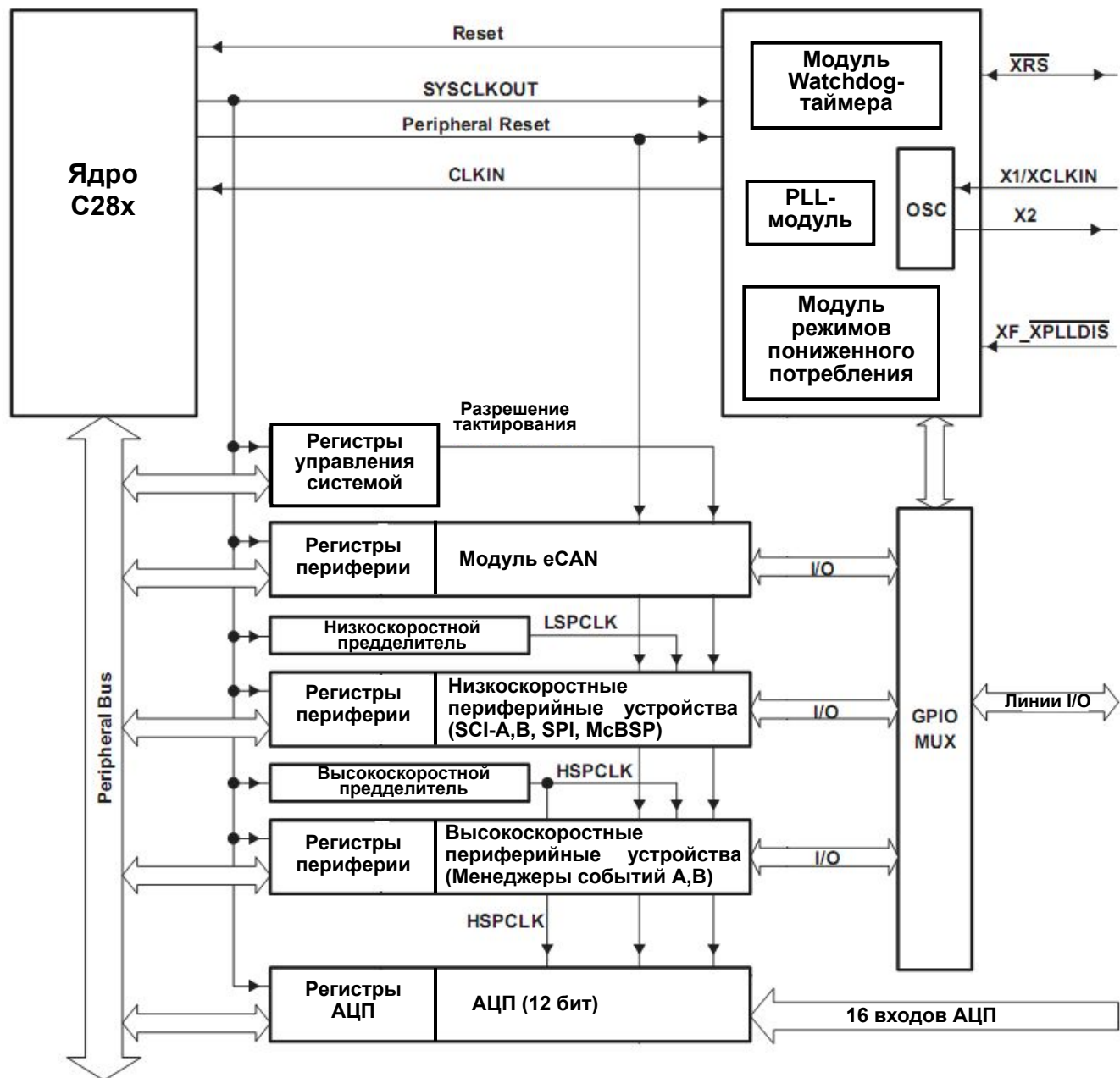
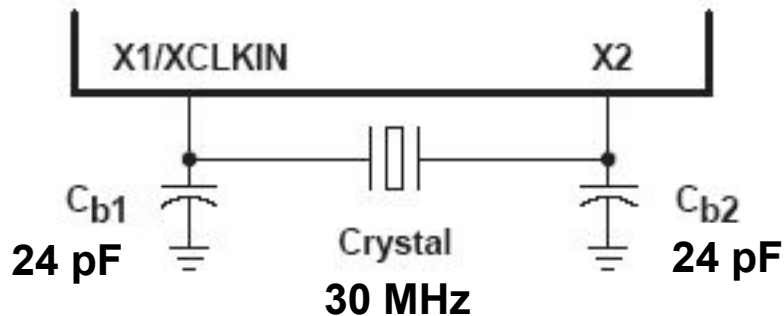


# Сброс и синхронизация ЦСП TMS320F28x



## Сброс и синхронизация ЦСП TMS320F28x

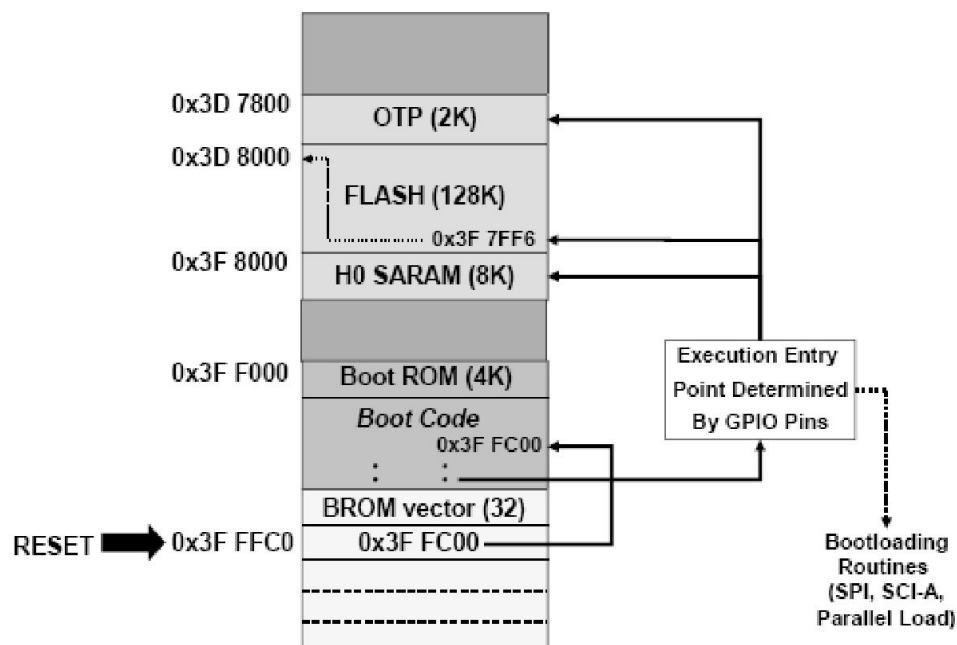
Как и большинство современных процессоров, TMS320F28x тактируется от генератора более низкой внешней частоты для уменьшения электромагнитных помех. Встроенный тактовый генератор позволяет подключить кварцевый резонатор к выводам X1/XCLKIN и X2. Если используется внешний генератор, он подключается к выводу X1/XCLKIN, а вывод X2 оставляют неподключенным:



# Сброс и синхронизация ЦСП TMS320F28x

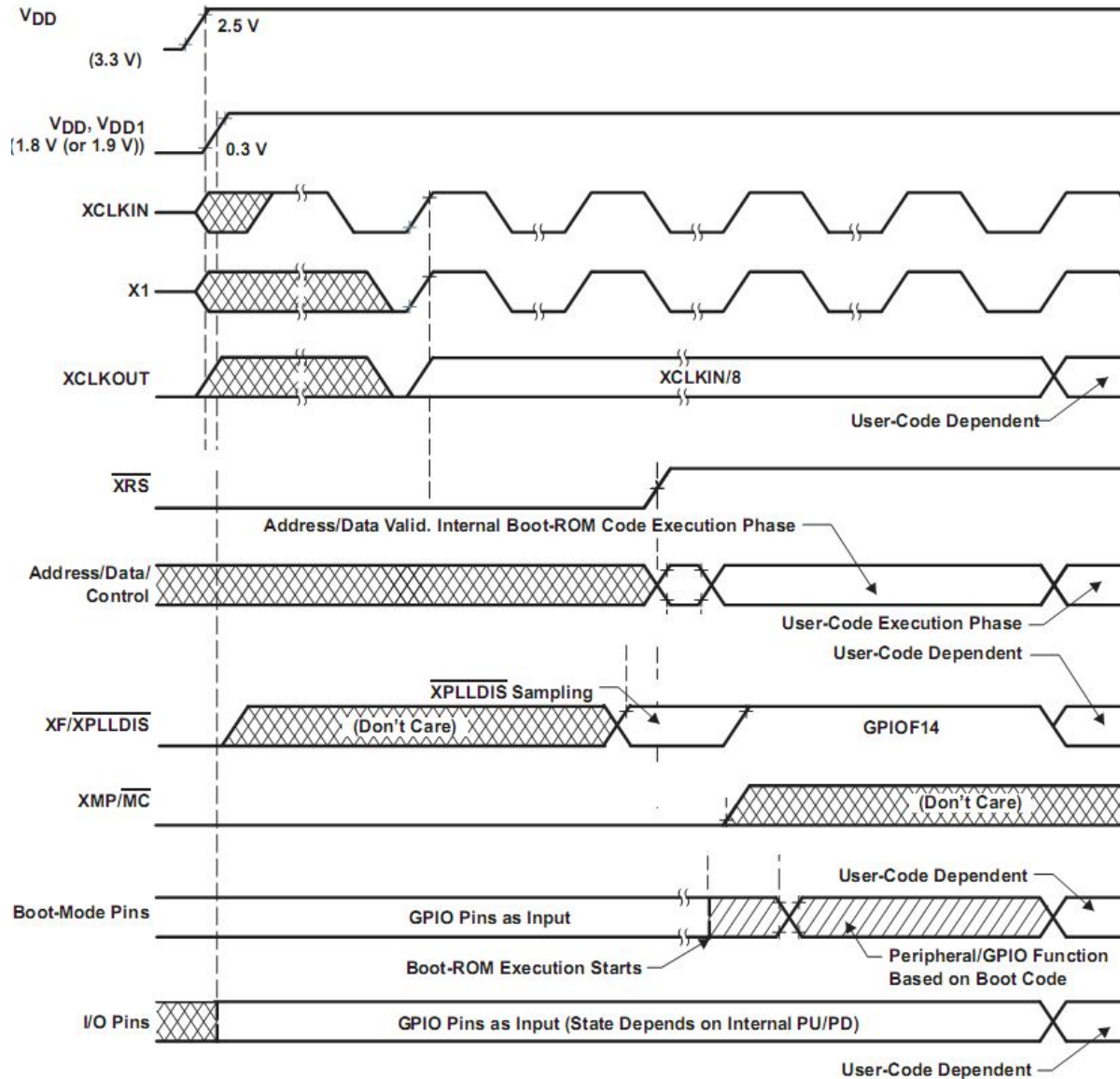
После сброса в режиме микроконтроллера запускается служебная программа Bootloader, которая анализирует выходы порта GPIOF (GPIOF2, GPIOF3, GPIOF4 и GPIOF12) и, исходя из комбинации сигналов на них, выполняет один из переходов:

Выводы GPIO				Режим запуска
F4	F12	F3	F2	
1	x	x	x	Передать управление FLASH-памяти по адресу 0x3F 7FF6
0	0	1	0	Передать управление H0 SARAM-памяти по адресу 0x3F 8000
0	0	0	1	Передать управление OTP-памяти по адресу 0x3D 7800
0	1	x	x	Загрузить программу из внешнего EEPROM во внутреннюю память через SPI-порт
0	0	1	1	Загрузить программу во внутреннюю память через SCI-A порт
0	0	0	0	Загрузить программу во внутреннюю память через параллельный порт GPIOB



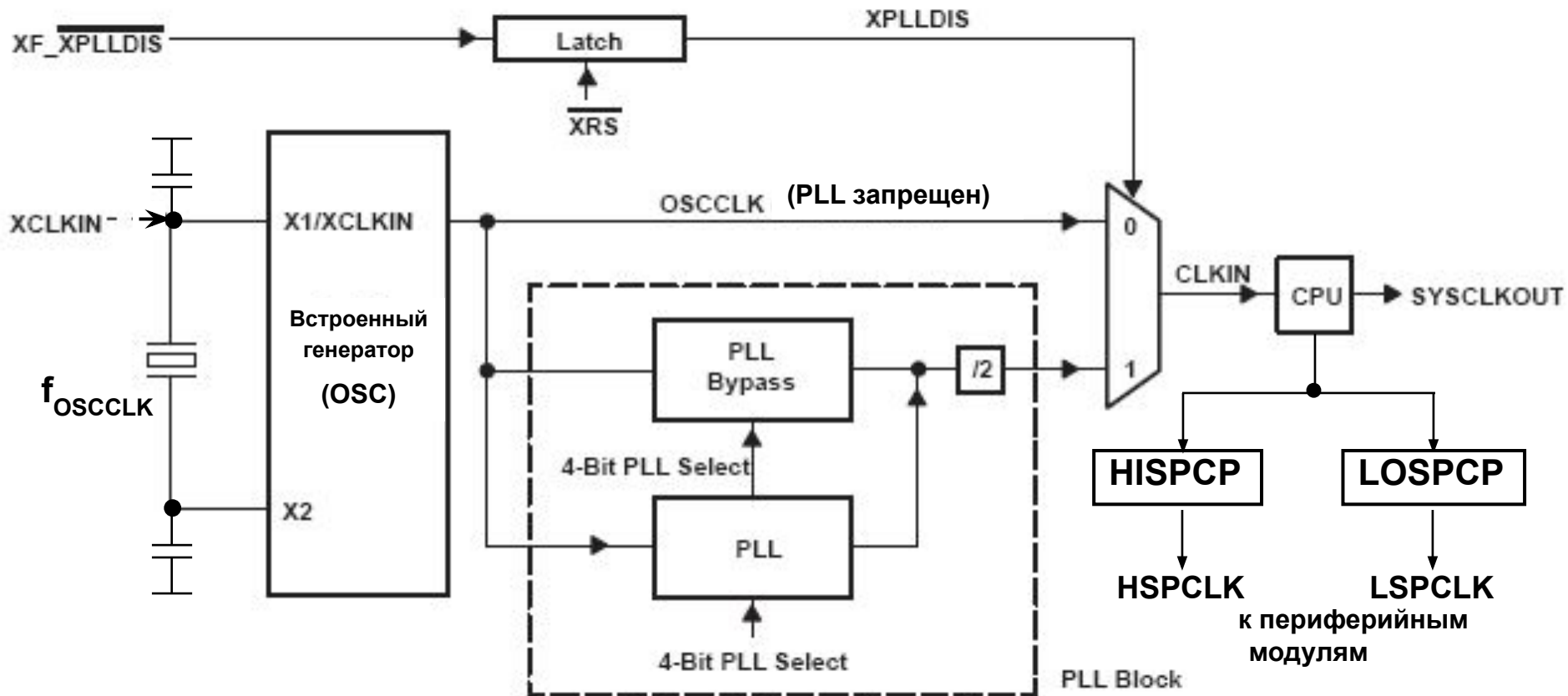
# Сброс и синхронизация ЦСП TMS320F28x

Временная диаграмма сигналов сброса и синхронизации:



# Сброс и синхронизация ЦСП TMS320F28x

## Внутренние сигналы синхронизации DSP TMS320F2812:



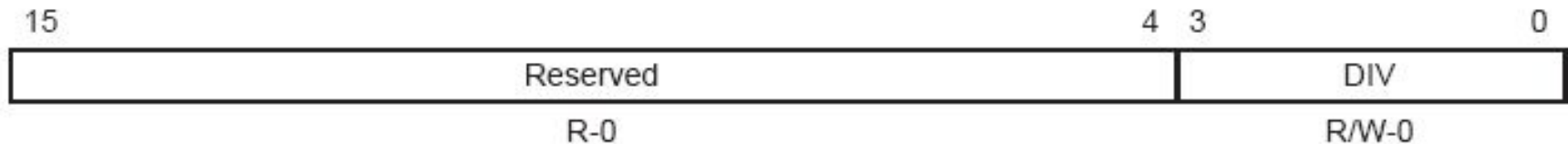
# Сброс и синхронизация ЦСП TMS320F28x

Модуль фазовой синхронизации (PLL – Phase-Locked Loop) задает внутреннюю частоту DSP. Возможные режимы:

**1. PLL-disabled (PLL запрещен).** Задается установкой вывода XPLLDIS в «0» в момент системного сброса (сигнал XRS). Частота  $f_{CLKIN}$  совпадает с входной тактовой частотой  $f_{OSCCLK}$ .

**2. PLL-bypass.**  $f_{CLKIN} = f_{OSCCLK}/2$  (режим после начальной установки, если PLL не запрещен).

**3. PLL-enabled (PLL разрешен).** Задается записью в регистр PLLCR ненулевого значения. Коэффициент умножения/деления частоты  $f_{CLKIN}$  задается в регистре PLLCR.



Частота CLKIN, на которой работает CPU (совпадает с частотой SYSCLKOUT тактирования периферии на выходе CPU) устанавливается в соответствии с четырехбитным кодом DIV0...DIV3.

# Сброс и синхронизация ЦСП TMS320F28x

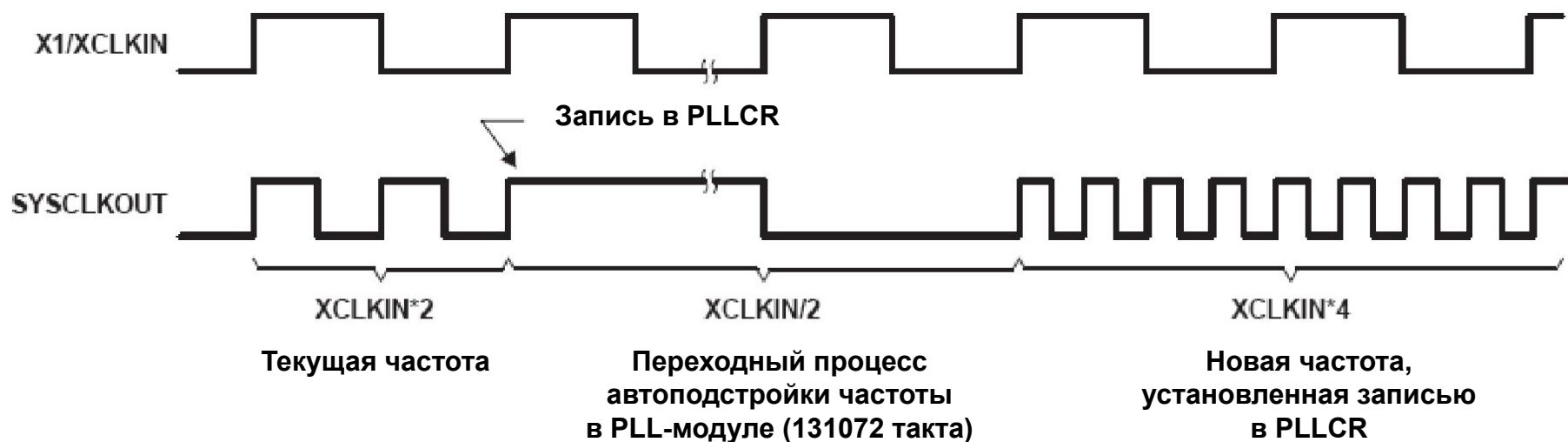
Назначение информационных битов в регистре PLLCR:

<i>DIV3</i>	<i>DIV2</i>	<i>DIV1</i>	<i>DIV0</i>	<i>Тактовая частота (SYSCLKOUT)</i>
0	0	0	0	XCLKIN/2 (PLL bypass)
0	0	0	1	XCLKIN/2
0	0	1	0	XCLKIN
0	0	1	1	XCLKIN*1,5
0	1	0	0	XCLKIN*2
0	1	0	1	XCLKIN*2,5
0	1	1	0	XCLKIN*3
0	1	1	1	XCLKIN*3,5
1	0	0	0	XCLKIN*4
1	0	0	1	XCLKIN*4,5
1	0	1	0	XCLKIN*5

Кроме состояния «PLL bypass»,  $SYSCLKOUT = (XCLKIN * n) / 2$ , где *n*-код в регистре PLLCR.

# Сброс и синхронизация ЦСП TMS320F28x

## Эффект записи в регистр PLLCR:



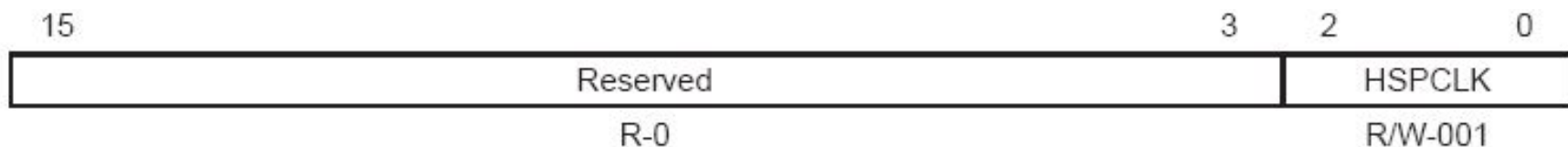
Время переключения на новую частоту составляет 131072 периодов входной тактовой частоты (в течение этого времени устанавливается системная частота  $f_{SYSCLKOUT} = f_{CLKIN}/2$ ), поэтому в программах иногда после записи в регистр PLLCR, необходимо вставлять программную временную задержку.



## Сброс и синхронизация ЦСП TMS320F28x

Регистры HISPCP (высокоскоростной предделитель) и LOSPCP (низкоскоростной предделитель) предназначены для конфигурации тактирования различных периферийных модулей. Эти регистры имеют аналогичный формат:

### HISPCP:



### LOSPCP:



# Сброс и синхронизация ЦСП TMS320F28x

Назначение информационных битов в регистрах HISPCP и LOSPCP :

<i>H/LSPCLK2</i>	<i>H/LSPCLK1</i>	<i>H/LSPCLK0</i>	<i>Тактовая частота периферии</i>
0	0	0	SYSCCLKOUT/1
0	0	1	SYSCCLKOUT/2 (по умолч. HISPCP)
0	1	0	SYSCCLKOUT/4 (по умолч. LOSPCP)
0	1	1	SYSCCLKOUT/6
1	0	0	SYSCCLKOUT/8
1	0	1	SYSCCLKOUT/10
1	1	0	SYSCCLKOUT/12
1	1	1	SYSCCLKOUT/14

Кроме состояния «все нули»,  $f_{\text{такт.}} = \text{SYSCCLKOUT}/(2*n)$ , где n- код в регистре HISPCP (LOSPCP).

# Сброс и синхронизация ЦСП TMS320F28x

Тактированием различных периферийных модулей управляет регистр PCLKCR:

15	14	13	12	11	10	9	8
Reserved	ECANENCLK	Reserved	MCBSPENCLK	SCIBENCLK	SCIAENCLK	Reserved	SPIENCLK
R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
7	4			3	2	1	0
Reserved				ADCENCLK	Reserved	EVBENCLK	EVAENCLK
R-0				R/W-0	R-0	R/W-0	R/W-0

# Сброс и синхронизация ЦСП TMS320F28x

Назначение информационных битов регистра PCLKCR :

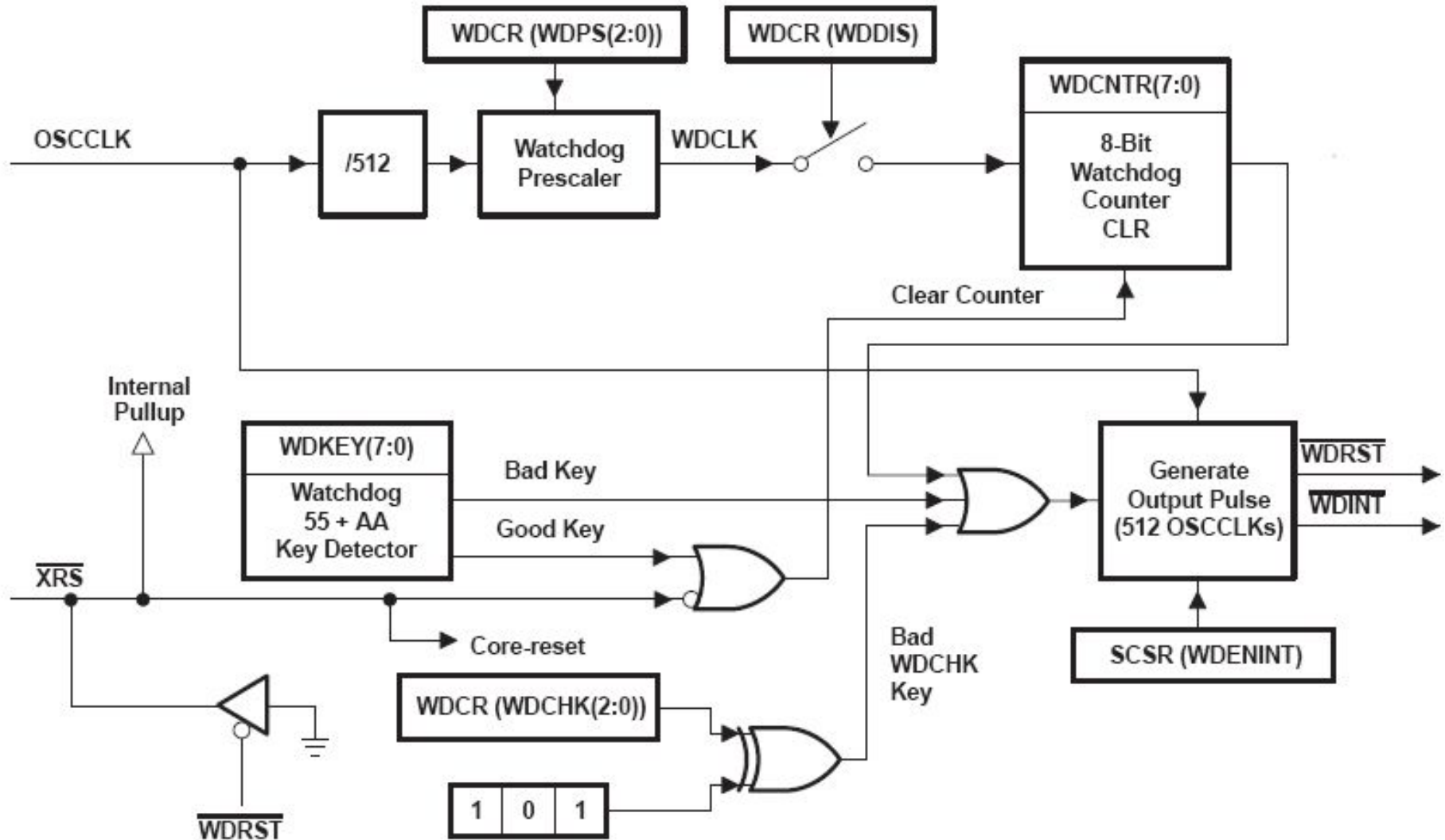
<i>Название</i>	<i>Назначение</i>
<b>EVAENCLK</b>	<b>Разрешение HSPCLK в EVA</b>
<b>EVBENCLK</b>	<b>Разрешение HSPCLK в EVB</b>
<b>ADCENCLK</b>	<b>Разрешение HSPCLK в ADC</b>
<b>SPIAENCLK</b>	<b>Разрешение LSPCLK в SPI</b>
<b>SCIAENCLK</b>	<b>Разрешение LSPCLK в SCI-A</b>
<b>SCIBENCLK</b>	<b>Разрешение LSPCLK в SCI-B</b>
<b>MCBSPENCLK</b>	<b>Разрешение LSPCLK в McBSP</b>
<b>ECANENCLK</b>	<b>Разрешение тактирования eCAN</b>

## Сброс и синхронизация ЦСП TMS320F28x

Сторожевой таймер (Watchdog-timer, WDT) – это счетное устройство, которое периодически вырабатывает сигнал сброса процессора, предотвращая таким образом его возможное зависание. Модуль Watchdog-таймера формирует выходной импульс сброса/прерывания WDRST / WDINT длительностью 512 периодов входной тактовой частоты в случае переполнения 8-битного счетчика WDCNTR.

# Сброс и синхронизация ЦСП TMS320F28x

Функциональная схема модуля WDT:



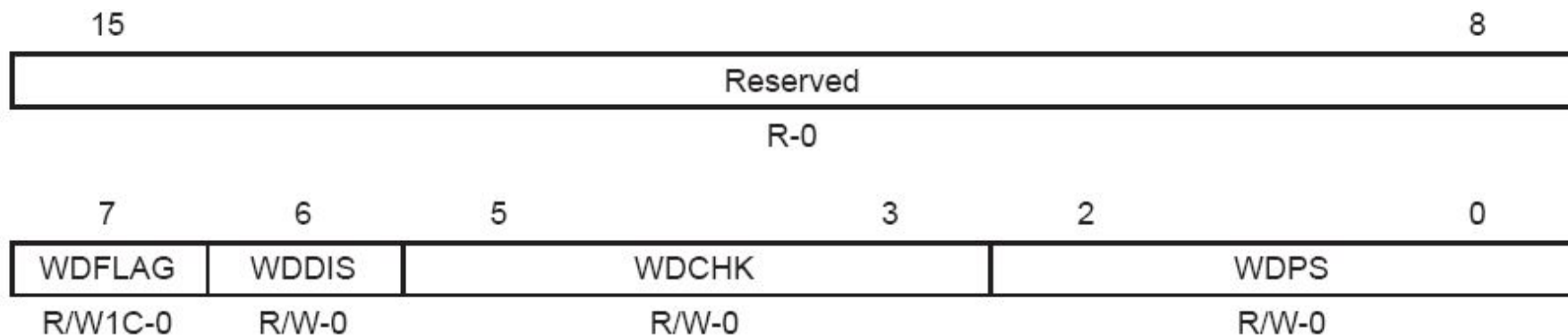
## Сброс и синхронизация ЦСП TMS320F28x

Для предотвращения сброса процессора необходимо периодически программно сбрасывать счетчик WDCNTR при помощи записи последовательности кодов «0x55 + 0xAA» в специальный регистр WDKKEY. При этом запись «0x55» разрешает сброс счетчика WDCNTR, а последующая запись «0xAA» сбрасывает этот счетчик. При этом сброса/прерывания ЦСП от WDT не происходит. Целесообразнее всего периодически выполнять последовательную запись значений «0x55» и «0xAA» в регистр WDKKEY в подпрограммах обслуживания прерывания (ISR) от системных таймеров. Запись любого иного значения в регистр WDKKEY приведет к немедленному сбросу ЦСП.

WDT всегда активируется при включении питания. Единственный способ обойтись без программного обслуживания WDT – это его отключение посредством установки 6-го бита (WDDIS) регистра управления сторожевого таймера (WDCR) в единицу.

# Сброс и синхронизация ЦСП TMS320F28x

Формат регистра управления сторожевого таймера WDCR :



**Legend:** R = Read access, W = write access, W1C = write 1 to clear, -0 = value after reset



# Сброс и синхронизация ЦСП TMS320F28x

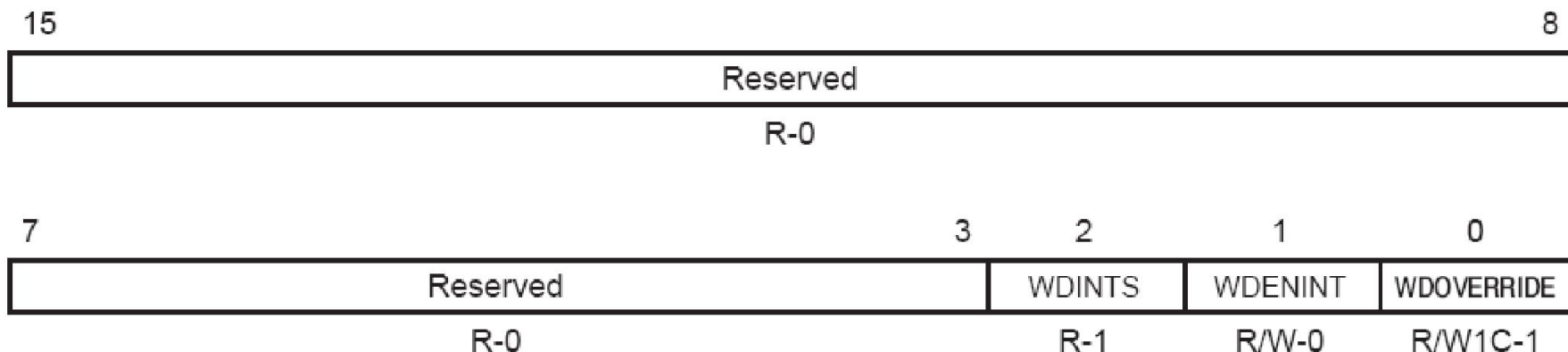
Назначение информационных битов регистра WDCR :

<i>Название</i>	<i>Назначение</i>
<b>WDFLAG</b>	Используется для отображения типа сброса процессора: обычный сброс (WDFLAG = 0) или сброс по сторожевому таймеру (WDFLAG = 1). Программный сброс этого бита – записью «1». Запись «0» игнорируется.
<b>WDDIS</b>	Запрет WDT (записью «1»)
<b>WDCNK(2-0)</b>	Биты защиты. Они должны иметь комбинацию «101», иначе произойдет немедленный сброс процессора
<b>WDPS(2-0)</b>	Выбор необходимого коэффициента деления частоты WDCLK для работы сторожевого таймера: 000 – WDCLK=OSCCLK/512/1 001 – WDCLK=OSCCLK/512/1 010 – WDCLK=OSCCLK/512/2 011 – WDCLK=OSCCLK/512/4 100 – WDCLK=OSCCLK/512/8 101 – WDCLK=OSCCLK/512/16 110 – WDCLK=OSCCLK/512/32 111 – WDCLK=OSCCLK/512/64

# Сброс и синхронизация ЦСП TMS320F28x

Регистр управления системой и статуса SCSR содержит бит разрешения блокировки Watchdog-таймера и биты управления прерыванием от Watchdog-таймера.

## Формат регистра SCSR:



# Сброс и синхронизация ЦСП TMS320F28x

Назначение информационных битов регистра SCSR:

<i>Название</i>	<i>Назначение</i>
<b>WDINTS</b>	Бит-индикатор прерывания от WDT. Отражает текущее состояние сигнала <u>WDINT</u> .
<b>WDENINT</b>	Разрешение сброса/прерывания от WDT. Если <b>WDENINT</b> = 0 (состояние по умолчанию после сброса ЦСП) – разрешен сигнал сброса <u>WDRST</u> . Если <b>WDENINT</b> = 1 – разрешен сигнал прерывания <u>WDINT</u> .
<b>WDOVERRIDE</b>	Если бит <b>WDOVERRIDE</b> установлен в 1, можно запрещать или разрешать работу WDT (битом <b>WDDIS</b> в регистре управления <b>WDCR</b> ). Сброс бита <b>WDOVERRIDE</b> осуществляется записью в него «1».

# Режимы пониженного потребления ЦСП TMS320F28x

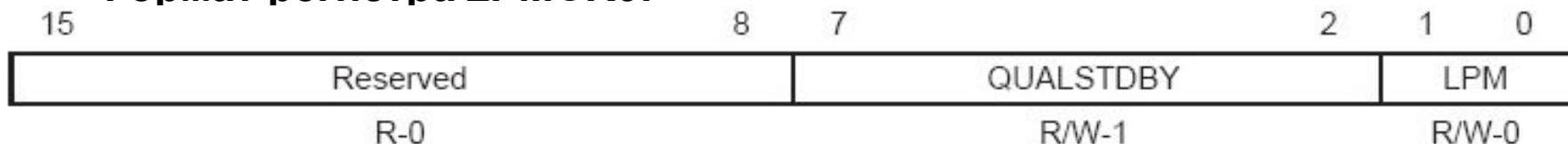
ЦСП TMS320F2812 может быть переведен в один из трех режимов пониженного потребления:

<i>Режим</i>	<i>Описание</i>	<i>OSCCLK</i>	<i>CLKIN</i>	<i>SYSCLKOUT</i>
<b>IDLE</b>	Переводит процессор в режим пониженного потребления. Тактирование периферии может быть избирательно заблокировано. Выход – любым разрешенным прерыванием тактируемой периферии, сигналами $\overline{XRS}$ , WAKEINT, внешним прерыванием XNMI_XINT13	On	On	On
<b>STANDBY</b>	Отключает процессор и периферию. Сохраняет работу тактового генератора и PLL-модуля. Выход – сигналами $\overline{XRS}$ , WAKEINT, XINT1, XNMI_XINT13, прерываниями периферии	On (WDT работает)	Off	Off
<b>HALT</b>	Отключает тактовый генератор (режим наименьшего возможного потребления). Выход – только сигналом $\overline{XRS}$ и внешним прерыванием XNMI_XINT13	Off	Off	Off

# Режимы пониженного потребления ЦСП TMS320F28x

Режимы пониженного потребления конфигурируются при помощи регистров LPMCR0 и LPMCR1.

Формат регистра LPMCR0:



Назначение информационных битов регистра LPMCR0 :

<i>Название</i>	<i>Назначение</i>
LPM(1-0)	<p>Выбор режима пониженного потребления (происходит после выполнения команды «IDLE»):</p> <p>00 – режим IDLE                      01 – режим STANDBY                      1x – режим HALT</p>
QUALSTDBY(5-0)	<p>Эти биты указывают минимальную длительность (в тактах входной частоты) сигнала, по которому производится выход из режима STANDBY пониженного потребления:</p> <p>000000 = 2 T<sub>OSCCLKs</sub>                      000001 = 3 T<sub>OSCCLKs</sub>                      ...                      111111 = 65 T<sub>OSCCLKs</sub></p>

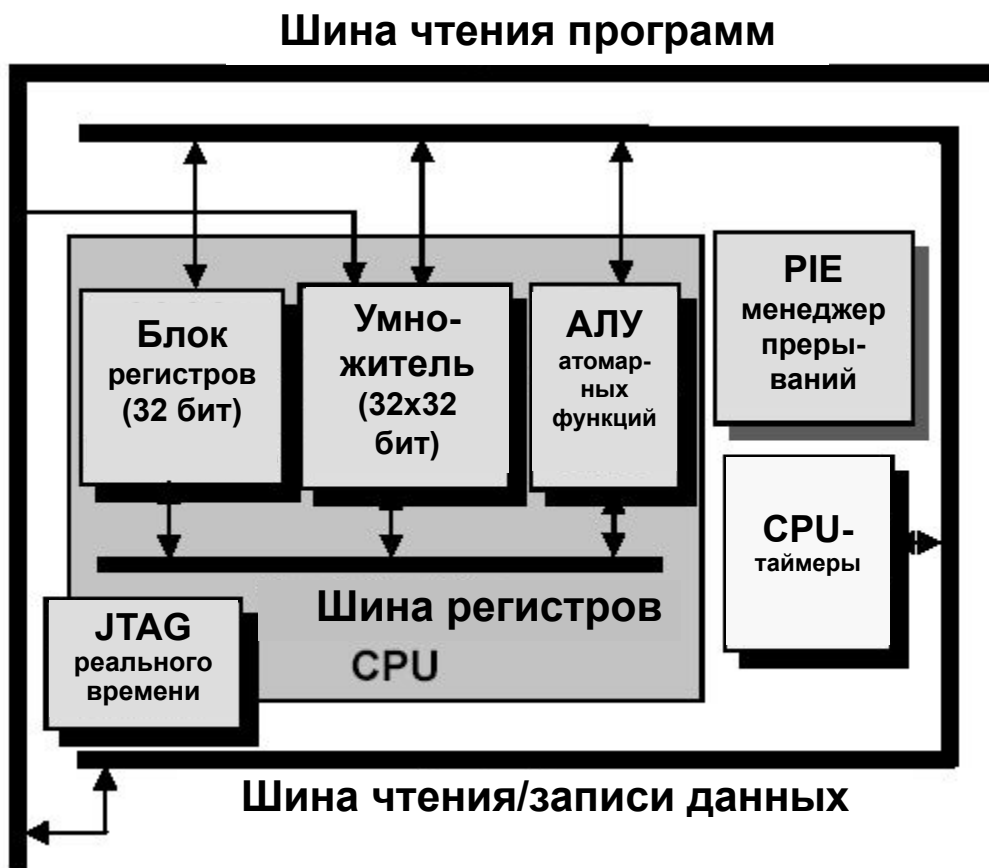
# Режимы пониженного потребления ЦСП TMS320F28x

## Формат регистра LPMCR1:

15	14	13	12	11	10	9	8
CANRX	SCIRXB	SCIRXA	C6TRIP	C5TRIP	C4TRIP	C3TRIP	C2TRIP
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
C1TRIP	T4CTRIP	T3CTRIP	T2CTRIP	T1CTRIP	WDINT	XNMI	XINT1
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

При установке в 1 соответствующего бита разрешается выход ЦСП из **STANDBY**-режима пониженного потребления посредством подачи соответствующего периферийного сигнала на один внешних выводов процессора.

# Модуль центрального процессора TMS320F28x



Модуль центрального процессора содержит:

- 32-битный RISC-процессор с фиксированной точкой;
- 32-битное атомарное АЛУ, выполняющее одноклеточные инструкции «чтение-модификация-запись»;
- блок 32-битных регистров;
- отладчик реального времени JTAG;
- систему шин, обеспечивающую гарвардскую архитектуру;
- три 32-битных таймера (CPU Timers);
- контроллер расширения прерываний (PIE);
- аппаратный модуль MAC 32 x 32 бит или двойной модуль MAC (DMAC) 16 x 16 с фиксированной точкой;
- аппаратное сдвигающее устройство

## Модуль центрального процессора TMS320F28x

Аппаратный умножитель 32x32 бит и 32-битное АЛУ обеспечивают совместное выполнение операций умножения и суммирования. Результат умножения может быть представлен 64-битным числом, операнды могут быть знаковыми и беззнаковыми. Большинство инструкций выполняется за один машинный цикл.

Блок 32-битных регистров (32-bit Auxiliary Registers – XAR0...XAR7) снабжен арифметическим блоком адресации регистров (ARAU – Address register arithmetic unit), который может инкрементировать и декрементировать регистры XAR0...XAR7 и SP и генерирует адреса памяти данных для записи (на шину DWAB) и чтения (на шину DRAB).

Модуль центрального процессора содержит также связанное с умножителем и АЛУ аппаратное сдвигающее устройство (Barrel Shifter) с возможностью сдвига содержимого на значение от 16 бит влево до 16 бит вправо.



# Модуль центрального процессора TMS320F28x

## Регистры модуля центрального процессора

Модуль центрального процессора содержит регистры данных, математических операций, управления системой, которые не имеют адресов в общем адресном пространстве памяти.

<i>Обозначение регистра</i>	<i>Название регистра</i>	<i>Начальное значение</i>
<b>ACC, 32 бит</b>	<b>Аккумулятор</b>	<b>0x00000000</b>
<b>АН, 16 бит</b>	<b>Старшие 16 бит ACC</b>	<b>0x0000</b>
<b>AL, 16 бит</b>	<b>Младшие 16 бит ACC</b>	<b>0x0000</b>
<b>XAR0, 32 бит</b>	<b>Вспомогательный регистр 0</b>	<b>0x00000000</b>
<b>XAR1, 32 бит</b>	<b>Вспомогательный регистр 1</b>	<b>0x00000000</b>
<b>...</b>	<b>...</b>	<b>...</b>
<b>XAR7, 32 бит</b>	<b>Вспомогательный регистр 7</b>	<b>0x00000000</b>

## Модуль центрального процессора TMS320F28x

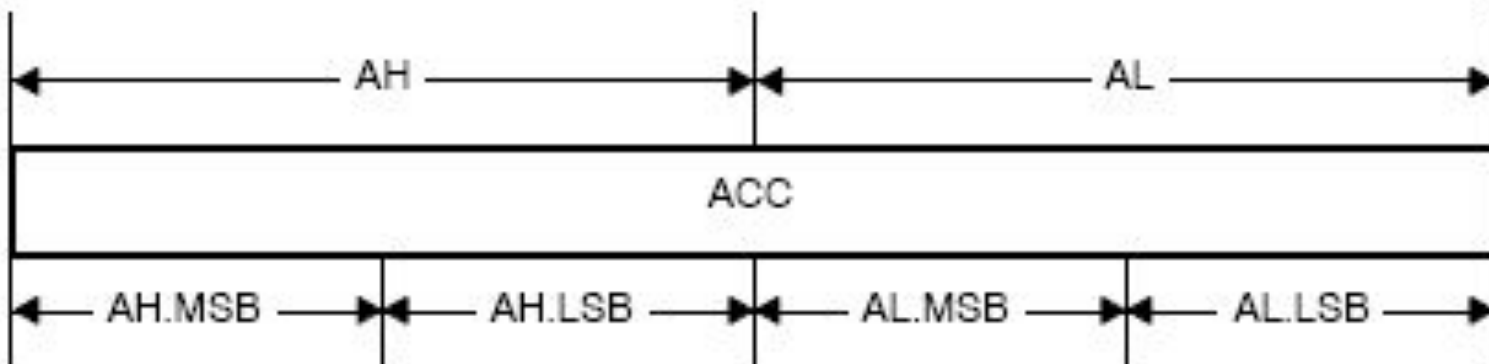
<i>Обозначение регистра</i>	<i>Название регистра</i>	<i>Начальное значение</i>
<b>AR0, 16 бит</b>	<b>Младшие 16 бит регистра XAR0</b>	<b>0x0000</b>
<b>AR1, 16 бит</b>	<b>Младшие 16 бит регистра XAR1</b>	<b>0x0000</b>
<b>...</b>	<b>...</b>	<b>...</b>
<b>AR7, 16 бит</b>	<b>Младшие 16 бит регистра XAR7</b>	<b>0x0000</b>
<b>DP, 16 бит</b>	<b>Указатель страницы данных</b>	<b>0x0000</b>
<b>IFR, 16 бит</b>	<b>Регистр флагов прерываний</b>	<b>0x0000</b>
<b>IER, 16 бит</b>	<b>Регистр разрешения прерываний</b>	<b>0x0000</b>
<b>DBGIER, 16 бит</b>	<b>Регистр разрешения отладки прерываний</b>	<b>0x0000</b>
<b>P, 32 бит</b>	<b>Регистр произведения</b>	<b>0x00000000</b>
<b>PH, 16 бит</b>	<b>Старшие 16 бит регистра P</b>	<b>0x0000</b>
<b>PL, 16 бит</b>	<b>Младшие 16 бит регистра P</b>	<b>0x0000</b>

## Модуль центрального процессора TMS320F28x

<i>Обозначение регистра</i>	<i>Название регистра</i>	<i>Начальное значение</i>
<b>PC, 22 бит</b>	<b>Программный счетчик</b>	<b>0x3FFFC0</b>
<b>RPC, 22 бит</b>	<b>Счетчик программного возврата</b>	<b>0x000000</b>
<b>SP, 16 бит</b>	<b>Указатель стека</b>	<b>0x0400</b>
<b>ST0, 16 бит</b>	<b>Регистр состояния 0</b>	<b>0x0000</b>
<b>ST1, 16 бит</b>	<b>Регистр состояния 1</b>	<b>0x080B</b>
<b>XT, 32 бит</b>	<b>Регистр множимого</b>	<b>0x00000000</b>
<b>T, 16 бит</b>	<b>Старшие 16 бит XT</b>	<b>0x0000</b>
<b>TL, 16 бит</b>	<b>Младшие 16 бит XT</b>	<b>0x0000</b>

## Модуль центрального процессора TMS320F28x

Аккумулятор (ACC) – главный рабочий регистр CPU. Это – адресат для всех операций АЛУ кроме тех, которые работают прямо в памяти или регистрах. ACC поддерживает операции сдвига, суммирования, вычитания и сравнения с 32-разрядными данными. Он может также принимать 32-разрядный результат операции умножения. Можно обращаться к половинам и четвертям ACC. ACC может быть обработан как два независимых 16-разрядных регистра: AH (старших 16 бит) и AL (младших 16 бит). К байтам в пределах AH и AL можно также обращаться независимо. Специальные команды загружают и сохраняют байты AH.MSB, AH.LSB, AL.MSB, AL.LSB Это позволяет эффективно упаковывать и распаковывать байты.



AH = ACC (31:16)

AH.MSB = ACC (31:24)

AH.LSB = ACC (23:16)

AL = ACC (15:0)

AL.MSB = ACC (15:8)

AL.LSB = ACC (7:0)

# Модуль центрального процессора TMS320F28x

Аккумулятор изменяет биты состояния регистра состояния ST0:

- бит режима расширения знака (SXM);
- бит режима переполнения (OVM);
- флаг тест/управление (TC);
- флаг переноса (C);
- флаг нуля (Z);
- флаг знака (N);
- запираемый флаг переполнения (V);
- биты счетчика переполнений (OVC, 6 бит).