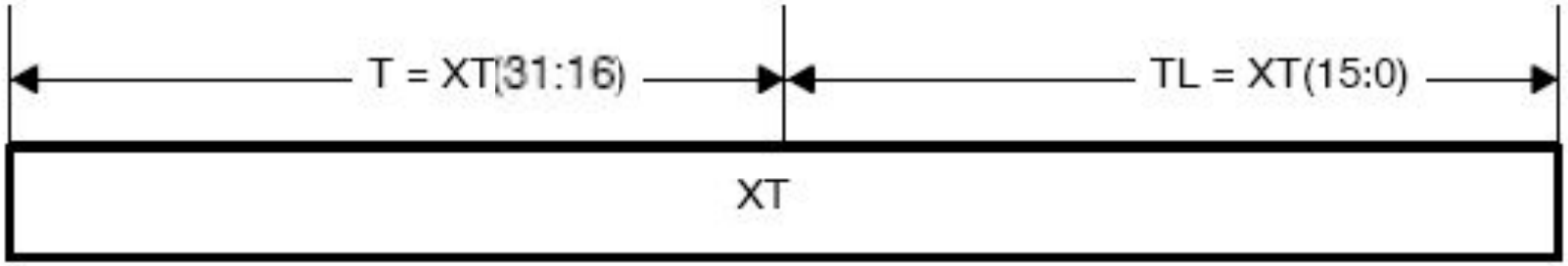
# Модуль центрального процессора TMS320F28x

**Регистр множимого (XT)** используется чаще всего для того чтобы сохранить 32-разрядное значение целого числа со знаком перед 32-битным умножением.

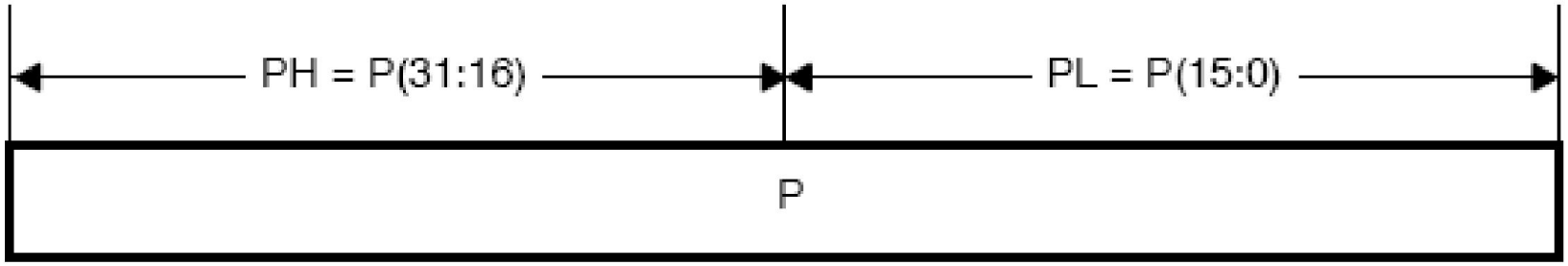
Младшая 16-разрядная часть регистра XT доступна как регистр TL, а старшая 16-разрядная часть – как регистр T.




Регистр T может использоваться, чтобы определить значение сдвига для некоторых операций сдвига влево и вправо. В этом случае, в зависимости от команды, используется от 4 до 6 младших бит регистра T, которые кодируют количество сдвигов – от 0 до 32, если сдвиг производится в аккумуляторе, и от 0 до 64, если сдвиг производится в паре регистров ACC и P.

# Модуль центрального процессора TMS320F28x

**Регистр произведения (P)** типично используется для захвата 32-разрядного результата умножения. Он может также быть загружен прямо 16- или 32-разрядными данными из ячейки памяти; 16-разрядной константой, 32-разрядным аккумулятором; 16-разрядным или 32-разрядным адресуемым регистром модуля CPU. Регистр P может быть обработан как 32-разрядный регистр или как два независимых 16-разрядных регистра: PH (старших 16 битов) и PL (младших 16 битов).



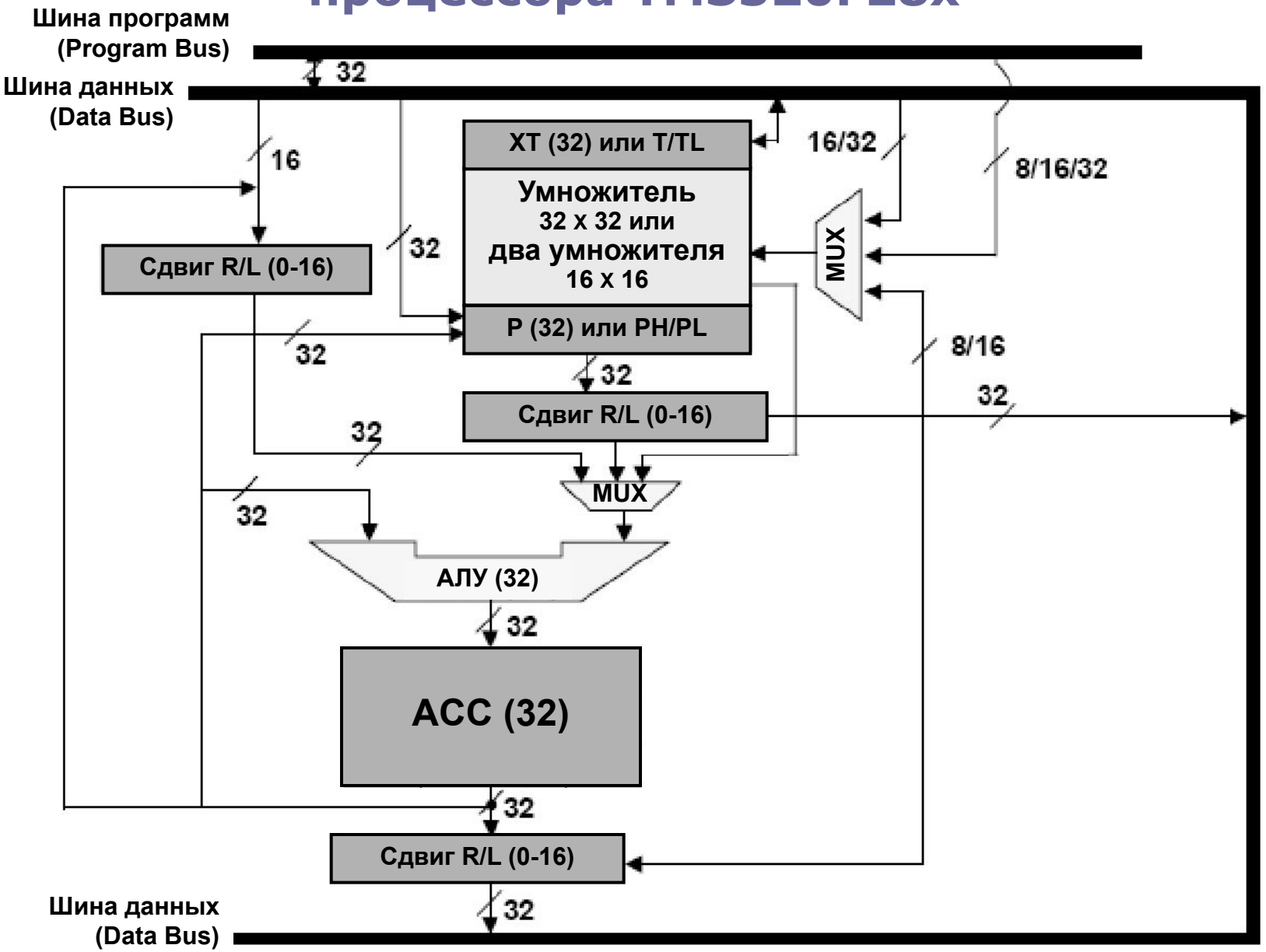
# Модуль центрального процессора TMS320F28x

**Когда определенные команды обращаются к P, PH, или PL, все 32 бита копируются в сдвигающее устройство АЛУ, которое может выполнять левый сдвиг, правый сдвиг или оставлять данные в исходном виде. Действие сдвигающего устройства для этих команд определено тремя битами режима сдвига (PM) регистра состояния ST0 и битом AMODE регистра состояния ST1:**

| *Значение битового поля PM* | *Режим сдвига* |
|---|---|
| 000 | Левый на 1 |
| 001 | Нет |
| 010 | Правый на 1 |
| 011 | Правый на 2 |
| 100 | Правый на 3 |
| 101 | Правый на 4 (если AMODE = 1, левый на 4) |
| 110 | Правый на 5 |
| 111 | Правый на 6 |

**Когда сдвигающее устройство исполняет левый сдвиг, младшие биты последовательно заполняются нулями. Команды, которые используют PH или PL как операнды, игнорируют режим сдвига.**

# Математический блок модуля центрального процессора TMS320F28x



# Математический блок модуля центрального процессора TMS320F28x

**При умножении регистр XT используется для захвата первого операнда и умножает его на второй, который загружается из памяти. Если регистр XT загружается из области памяти данных, а второй операнд извлекается из программной памяти, выполняется операция умножения за один машинный цикл. Результат умножения помещается в регистр произведения P или прямо в аккумулятор (ACC). При выполнении операции умножения 32x32 бит 64-разрядный результат помещается в регистры P и ACC.**

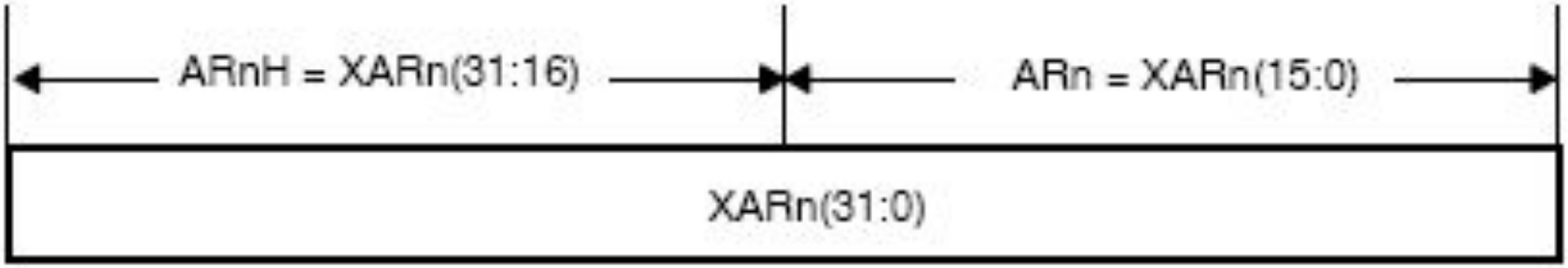
**Устройства аппаратного сдвига данных (Сдвиг R/L) могут быть использованы параллельно с другими аппаратными модулями CPU. Обычно они используются для масштабирования промежуточных результатов вычислений при реализации алгоритмов управления в реальном времени или для деления/умножения на $2^n$.**

**Арифметико-логическое устройство (АЛУ) выполняет операции статической (безсдвиговой) математики. При этом первый операнд всегда содержится в аккумуляторе (ACC, AH или AL). Второй операнд может быть загружен из памяти данных, памяти программ, из регистра P или непосредственно из умножителя.**

# Модуль центрального процессора TMS320F28x

**<u>Вспомогательные регистры (XAR0-XAR7).</u> CPU содержит восемь 32-разрядных регистров, которые могут использоваться как указатели ячеек памяти или как регистры общего назначения.**

**Многие команды позволяют обратиться к 16 младшим битам XAR0-XAR7. Эти 16 младших битов доступны как регистры AR0-AR7. AR0-AR7 могут использоваться как универсальные регистры для управления циклами и для 16-разрядных сравнений.**




**n – номер от 0 до 7**

# Модуль центрального процессора TMS320F28x

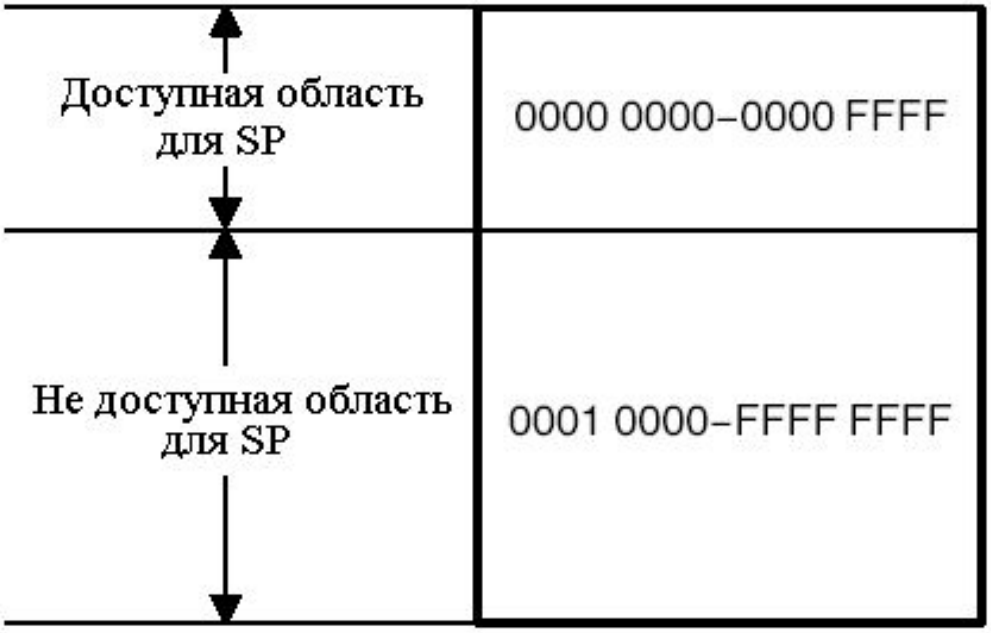
**При доступе AR0–AR7, старшие 16 битов регистра (обозначаемые как регистры AR0H–AR7H), могут изменяться или сохранять свои значения, в зависимости от используемой команды. Регистры AR0H–AR7H доступны только как часть регистров XAR0–XAR7 (индивидуально недоступны).**

**Для операций с ACC используются все 32 бита (@XARn). Для 16-разрядных операций, используются только младшие 16 битов, а старшие 16 битов игнорируются (@ARn).**

**XAR0–XAR7 могут также использоваться командами с косвенной адресацией, чтобы обратится к произвольному значению в памяти программ.**

# Модуль центрального процессора TMS320F28x

**Указатель вершины стека (SP)** разрешает использование программного стека в памяти данных. Указатель вершины стека имеет разрядность 16 бит и может адресовать только младшие 64Кб пространства данных. При сбросе SP указывает на адрес 0000 $0400_{16}$.



# Модуль центрального процессора TMS320F28x

**Стек увеличивается по направлению от младших адресов к старшим. Указатель стека всегда адресует следующие свободные ячейки. Когда 32-битное значение сохраняется в стеке, первыми по адресу ADR сохраняются младшие 16 бит, затем по адресу ADR+1 сохраняются старшие 16 бит. Если при выполнении вызовов подпрограмм, обработке прерываний, обработке команд PUSH и POP значение SP превышает $FFFF_{16}$, следующим его значением будет $0000_{16}$. Аналогично при уменьшении SP до значений меньше $0000_{16}$, следующим значением будет $FFFF_{16}$.**

# Модуль центрального процессора TMS320F28x

**Указатель страницы данных (DP).** **В режимах прямой адресации, память разбита на блоки (страницы данных) объемом по 64 слова. Младшие 4М слов памяти данных разбиты на 65536 блоков. В режиме прямой адресации 16-разрядный DP содержит текущий номер страницы данных. Для изменения страницы данных необходимо загрузить в DP новое значение.**

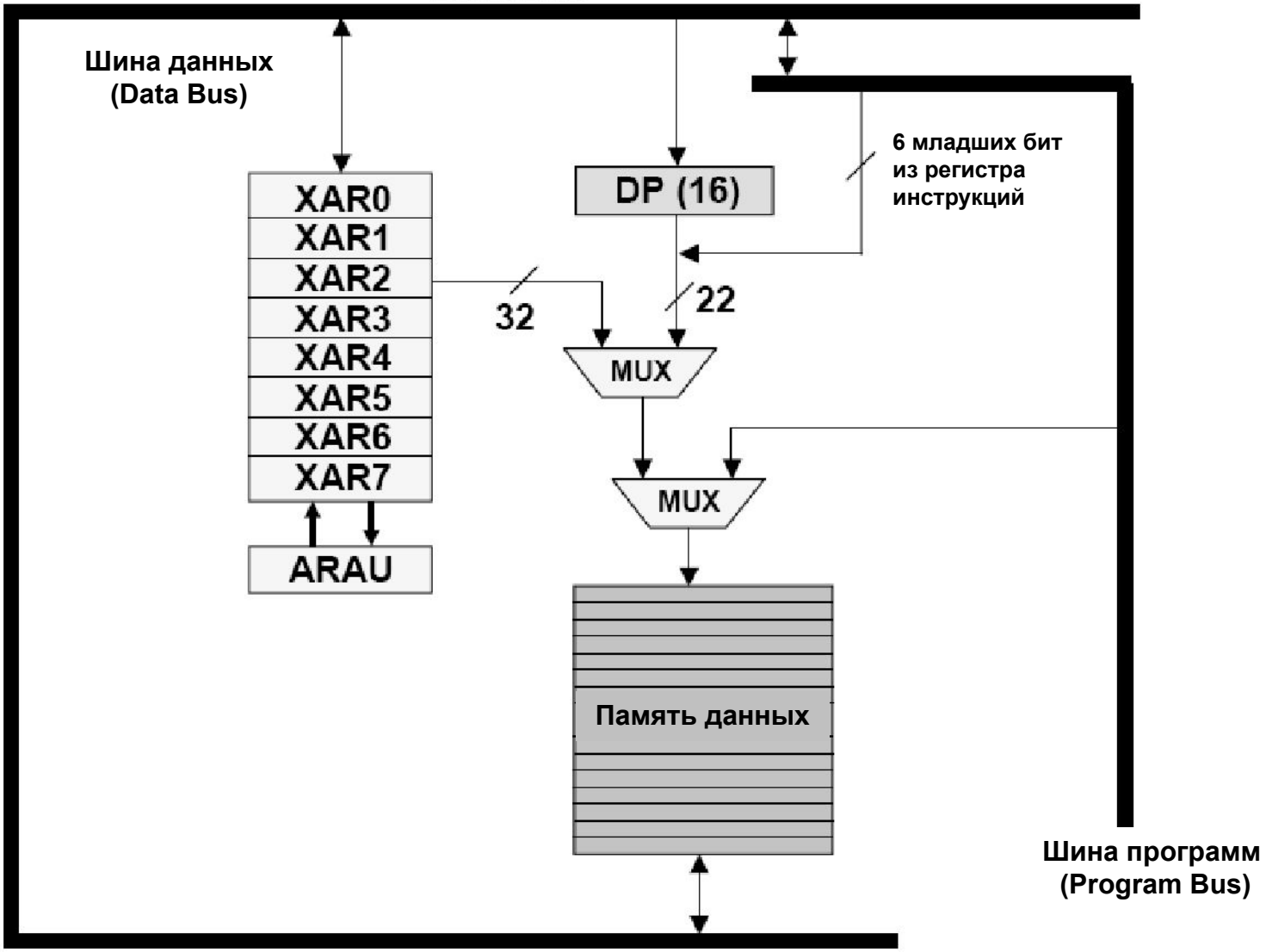
| Страница данных | Смещение | Область памяти данных |
|---|---|---|
| 00 0000 0000 0000 00<br>:<br>00 0000 0000 0000 00 | 00 0000<br>-<br>11 1111 | Страница 0: 0000 0000 – 0000 003F |
| 00 0000 0000 0000 01<br>:<br>00 0000 0000 0000 01 | 00 0000<br>-<br>11 1111 | Страница 1: 0000 0040 – 0000 007F |
| 00 0000 0000 0000 10<br>:<br>00 0000 0000 0000 10 | 00 0000<br>-<br>11 1111 | Страница 2: 0000 0080 – 0000 00BF |
| . | . | . |
| 11 1111 1111 1111 11<br>:<br>11 1111 1111 1111 11 | 00 0000<br>-<br>11 1111 | Страница 65535: 003F FFC0 – 003F FFFF |

# Блок адресации памяти данных модуля центрального процессора TMS320F28x

**Для доступа к произвольным областям памяти данных в ЦСП TMS320F2812 используются 2 основных режима:**

**- режим прямой адресации;**
**- режим косвенной адресации.**

# Блок адресации памяти данных модуля центрального процессора TMS320F28x



# Блок адресации памяти данных модуля центрального процессора TMS320F28x

**В режиме прямой адресации 22-битный адрес доступа к памяти генерируется путем соединения содержимого регистра DP (старшие 16 бит, Страница данных – Data Page) и 6 младших бит, задаваемых непосредственно в инструкции (Смещение – Offset).**

**Преимущество данного режима: когда DP установлен, имеется доступ к любой ячейке выбранной страницы в любом порядке.**

**Недостатки:**

**1) при необходимости доступа к данным другой страницы, предварительно необходимо изменить содержимое DP;**

**2) область памяти данных с адресами выше $003FFFFF_{16}$ (объем свыше 4M) недоступна с использованием регистра DP.**

# Блок адресации памяти данных модуля центрального процессора TMS320F28x

**В режиме косвенной адресации используются восемь 32-разрядных регистров XAR0…XAR7 для захвата полного 32-битного адреса операнда.**

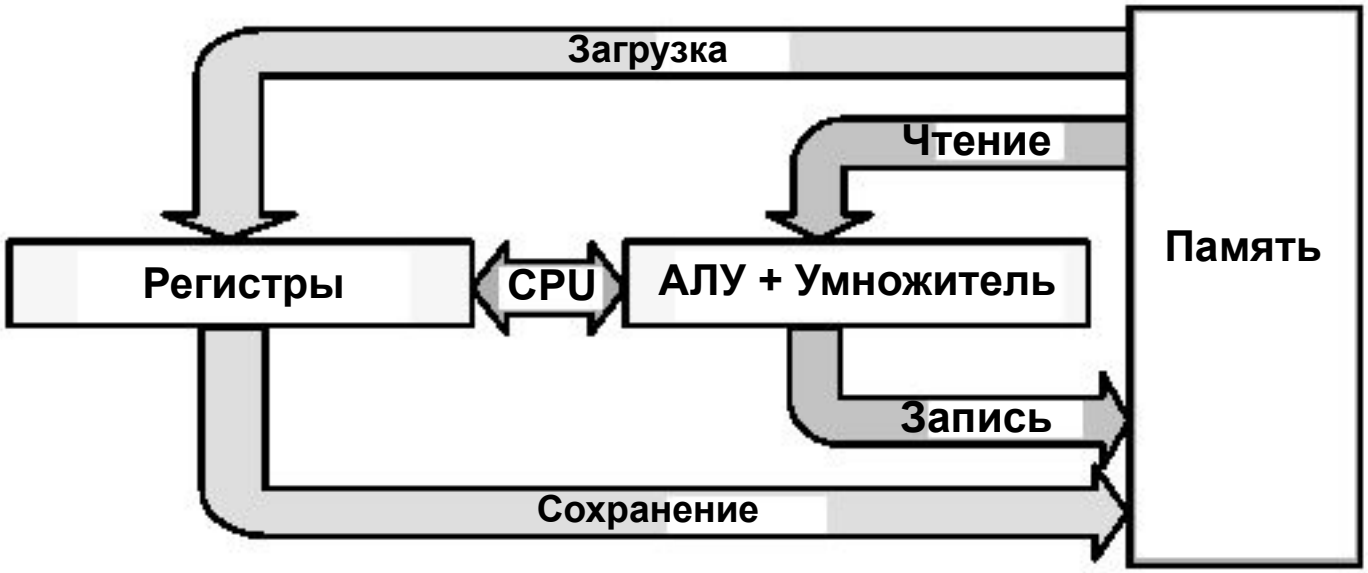
**Преимущество способа: при помощи ARAU (арифметического блока адресации регистров) «индексная» арифметика возможна в том же цикле, в котором осуществлялся доступ к ячейке памяти.**

**Недостаток: доступ к произвольной области памяти данных возможен только после новой установки регистра-указателя.**

**Режимы работы блока ARAU: постинкремент, преддекремент, индексное сложение и вычитание, стековые относительные операции, круговая и бит-реверсивная адресация.**

# Модуль центрального процессора TMS320F28x

**Атомарное АЛУ.** **Атомарными называются простые непрерываемые инструкции, в результате чего выполнение обработки данных по принципу «чтение – модификация – запись» может происходить на несколько циклов быстрее, чем по стандартному принципу «загрузка операнда – модификация – сохранение результата».**



# Модуль центрального процессора TMS320F28x

**Пример выполнения алгоритма без использования и с использованием атомарного АЛУ: ((XAR2))←((XAR2))&1234H.**

**Стандартное выполнение процедуры: загрузка – модификация - сохранение**

```
DINT
MOV    AL,*XAR2
AND    AL,#1234h
MOV    *XAR2,AL
EINT
```

**6 слов / 6 циклов**

**Выполнение процедуры с использованием атомарного АЛУ: чтение – модификация – запись**

```
AND   *XAR2,#1234h
```

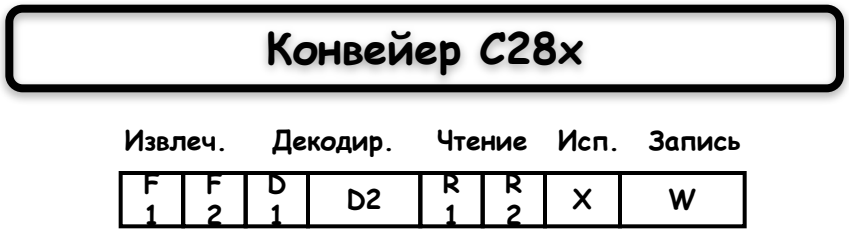
**2 слова / 1 цикл**

# Модуль центрального процессора TMS320F28x

***«Pipeline» - конвейерная структура выполнения команд***

**В ЦСП семейства C28x используется специальная 8-этапная структура выполнения команд для увеличения производительности.**

**Во время выполнения программы в восьми независимых фазах выполняются следующие базовые операции:**




**- Извлечение инструкций из памяти программ. Фазы Fetch1 (F1) и Fetch2 (F2);**

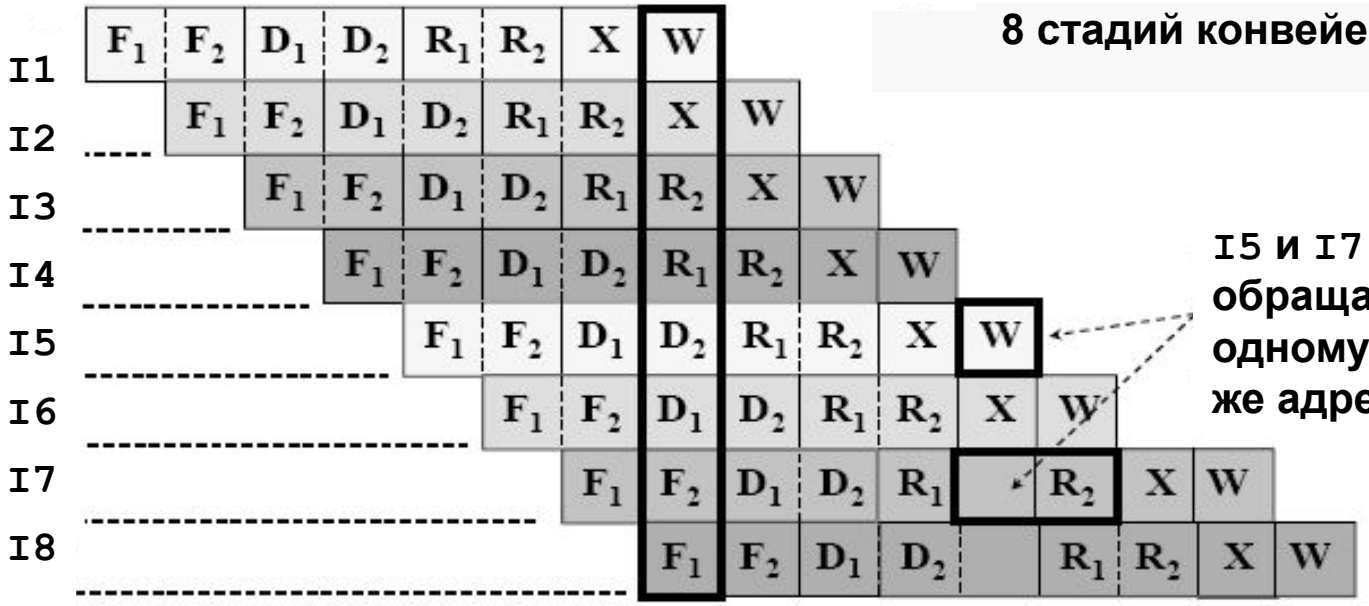
**- Декодирование инструкций. Фазы Decode1 (D1) и Decode2 (D2);**

**- Чтение данных из памяти или из регистров CPU. Фазы Read1 (R1) и Read2 (R2);**

**- Исполнение инструкций. Фаза eXecute (X);**

**- Запись результатов в память или регистры CPU. Фаза Write (W).**

# Модуль центрального процессора TMS320F28x




**F1:** Выставление 22-разрядного кода адреса инструкции на шину PAB;

**F2:** Извлечение кода инструкции через 32-разрядную шину PRDB и загрузка его в буфер инструкций;

**D1:** Декодирование инструкции. В этой же стадии распознаются границы текущей инструкции в буфере инструкций и определяется адрес следующей инструкции, которая будет выполняться;

**D2:** Определение адреса операнда. В этой же стадии инструкция из буфера инструкций загружается в регистр инструкций (IR), где завершается декодирование;

**R1:** Выставление адреса операнда на шину данных (DRAB) или шину программ (PAB);

**R2:** Чтение операнда через 32-разрядную шину (PRDB или DRDB);

**X:** Реальное действие (модификация операндов, содержимого регистров, изменение режима ЦСП и т.д.). Основные арифметические и логические операции выполняются в АЛУ, результат записывается в один из CPU-регистров в конце фазы;

**W:** Запись результата в память или регистры. CPU выставляет адрес приемника на шину DWAB, формирует соответствующий строб записи, и данные через шину DWDB записываются в память.

# Модуль центрального процессора TMS320F28x

| F1 | F2 | D1 | D2 | R1 | R2 | X | W | Циклы |
|---|---|---|---|---|---|---|---|---|
| I1 | | | | | | | | 1 |
| I2 | I1 | | | | | | | 2 |
| I3 | I2 | I1 | | | | | | 3 |
| I4 | I3 | I2 | I1 | | | | | 4 |
| I5 | I4 | I3 | I2 | I1 | | | | 5 |
| I6 | I5 | I4 | I3 | I2 | I1 | | | 6 |
| I7 | I6 | I5 | I4 | I3 | I2 | I1 | | 7 |
| I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | 8 |
| | I8 | I7 | I6 | I5 | I4 | I3 | I2 | 9 |
| | | I8 | I7 | I6 | I5 | I4 | I3 | 10 |
| | | | I8 | I7 | I6 | I5 | I4 | 11 |
| | | | | I8 | I7 | I6 | I5 | 12 |
| | | | | | I8 | I7 | I6 | 13 |
| | | | | | | I8 | I7 | 14 |
| | | | | | | | I8 | 15 |

# Модуль центрального процессора TMS320F28x

**F1: Выставление 22-разрядного кода адреса инструкции на шину PAB.**

**F2: Извлечение кода инструкции через 32-разрядную шину PRDB и загрузка его в буфер инструкций.**

**D1: Декодирование инструкции. В этой же стадии распознаются границы текущей инструкции в буфере инструкций и определяется адрес следующей инструкции, которая будет выполняться.**

**D2: Определение адреса операнда. В этой же стадии инструкция из буфера инструкций загружается в регистр инструкций (IR), где завершается декодирование.**

**R1: Выставление адреса операнда на шину данных (DRAB) или шину программ (PAB).**

**R2: Чтение операнда через 32-разрядную шину (PRDB или DRDB).**

**X: Реальное действие (модификация операндов, содержимого регистров, изменение режима ЦСП и т.д.). Основные арифметические и логические операции выполняются в АЛУ, результат записывается в один из CPU-регистров в конце фазы.**

**W: Запись результата в память или регистры. CPU выставляет адрес приемника на шину DWAB, формирует соответствующий строб записи, и данные через шину DWDB записываются в память.**