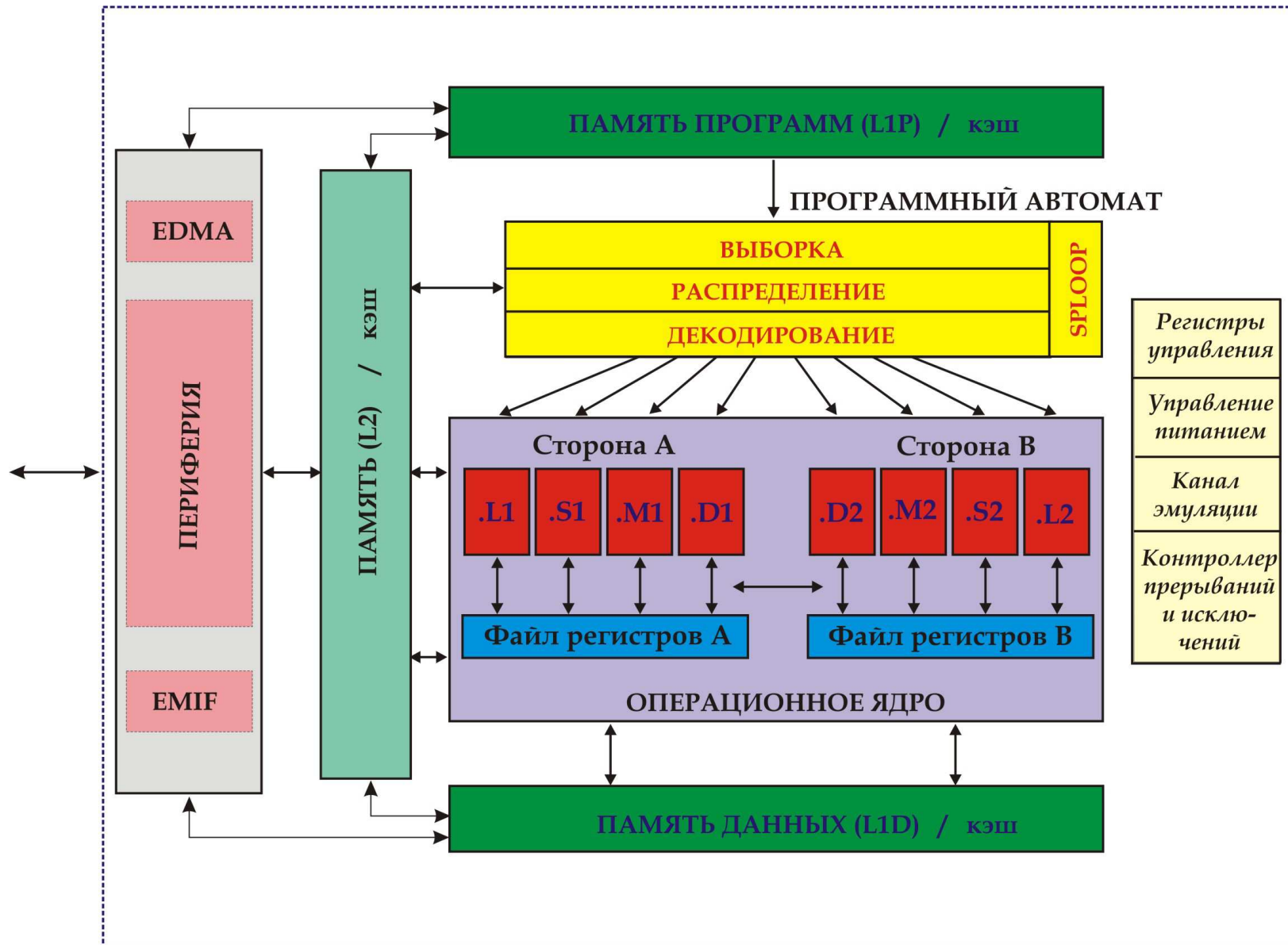




Архитектура цифровых сигнальных процессоров TMS320C674x *(продолжение)*



Архитектура ЦСП TMS320C674x: память





Память

Данные, участвующие в обработке, и коды программ хранятся в памяти процессора - **внутренней** (входит в состав процессора) или **внешней** (отдельные микросхемы).

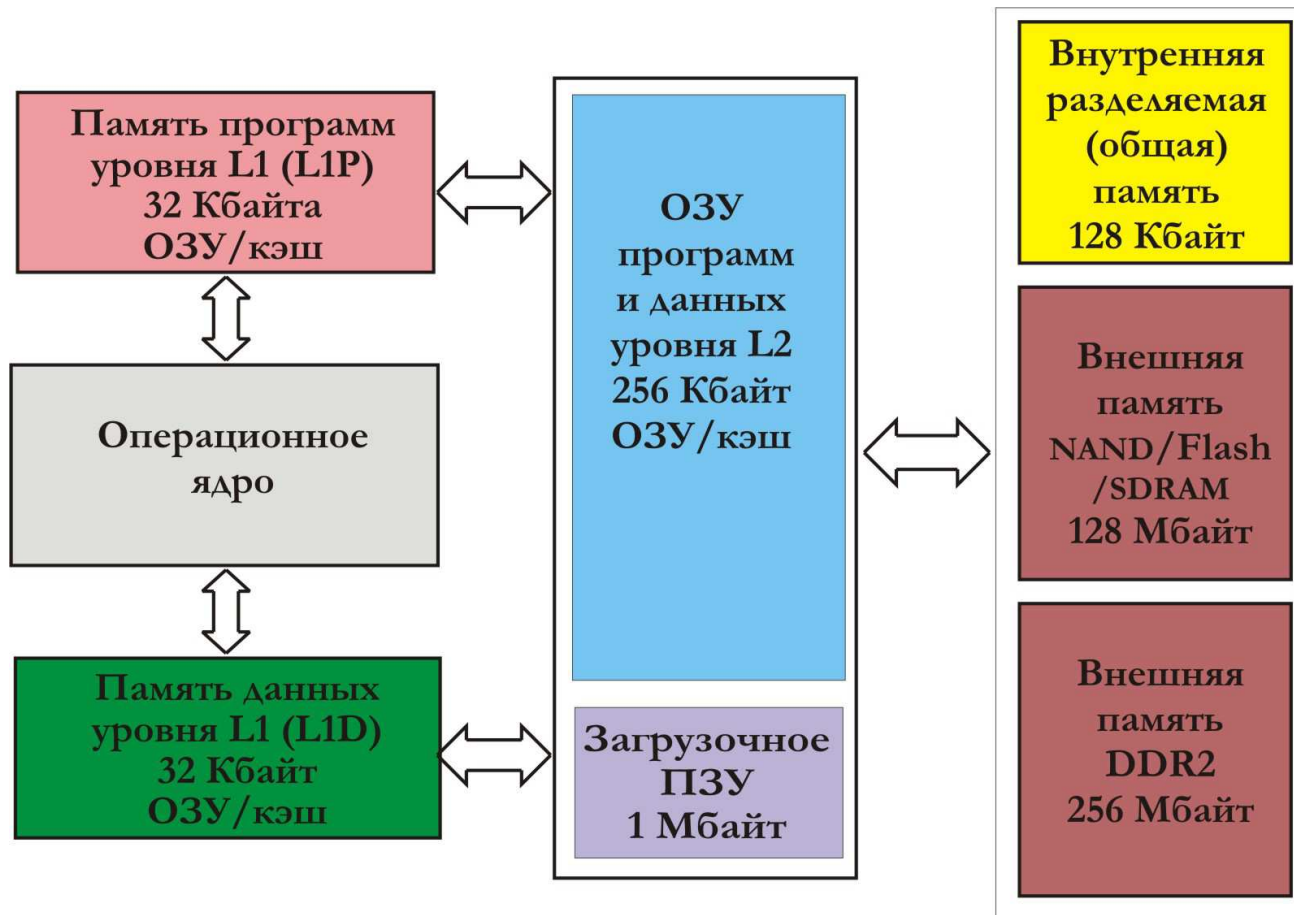
Обращение к требуемому операнду (команде), хранящемуся в памяти, происходит выставлением соответствующего адреса на **адресной шине**.

Адресная шина - 32-разрядная, что определяет общее адресное пространство процессора - 4 Гбайта.

В рамках этого общего пространства памяти организуется обращение к **внутренней памяти программ и данных**, различным типам **внешней памяти**, **внутренней памяти периферийных устройств**.



Структура памяти:





Для каждого из перечисленных типов памяти в общем адресном пространстве выделяется своя область со строго определенным диапазоном адресов. Графическое отображение расположения поддиапазонов памяти в общем адресном пространстве называется картой памяти (см. рисунок)

Адрес начала сегмента	Карта памяти TMS320C6748	Размер сегмента в байтах
0000 0000h	Регистры PRUSS	4 К
0070 0000h	Внутреннее ПЗУ L2	1024 К
0008 0000h	Внутреннее ОЗУ L2	256 К
000E 0000h	Внутреннее ОЗУ программ L1P	32 К
000F 0000h	Внутреннее ОЗУ данных L1D	32 К
0180 0000h	Регистры управления периферией	
4000 0000h	Внешняя память сегмент CS0 - CS5	640 М
C000 0000h	Внешняя память DDR2	256 М
FFFF FFFFh		



Карта памяти:

Table 3-4. C6748 Top Level Memory Map

Start Address	End Address	Size	DSP Mem Map	EDMA Mem Map	PRUSS Mem Map	Master Peripheral Mem Map	LCDC Mem Map
0x0000 0000	0x0000 0FFF	4K			PRUSS Local Address Space		
0x0000 1000	0x006F FFFF						
0x0070 0000	0x007F FFFF	1024K	DSP L2 ROM ⁽¹⁾				
0x0080 0000	0x0083 FFFF	256K	DSP L2 RAM				
0x0084 0000	0x00DF FFFF						
0x00E0 0000	0x00E0 7FFF	32K	DSP L1P RAM				
0x00E0 8000	0x00EF FFFF						
0x00F0 0000	0x00F0 7FFF	32K	DSP L1D RAM				
0x00F0 8000	0x017F FFFF						
0x0180 0000	0x0180 FFFF	64K	DSP Interrupt Controller				
0x0181 0000	0x0181 0FFF	4K	DSP Powerdown Controller				
0x0181 1000	0x0181 1FFF	4K	DSP Security ID				
0x0181 2000	0x0181 2FFF	4K	DSP Revision ID				
0x0181 3000	0x0181 FFFF	52K					
0x0182 0000	0x0182 FFFF	64K	DSP EMC				
0x0183 0000	0x0183 FFFF	64K	DSP Internal Reserved				
0x0184 0000	0x0184 FFFF	64K	DSP Memory System				
0x0185 0000	0x01BF FFFF						
0x01C0 0000	0x01C0 7FFF	32K		EDMA3 CC			
0x01C0 8000	0x01C0 83FF	1K		EDMA3 TC0			
0x01C0 8400	0x01C0 87FF	1K		EDMA3 TC1			
0x01C0 8800	0x01C0 FFFF						
0x01C1 0000	0x01C1 0FFF	4K		PSC 0			
0x01C1 1000	0x01C1 1FFF	4K		PLL Controller 0			
0x01C1 2000	0x01C1 3FFF						
0x01C1 4000	0x01C1 4FFF	4K		SYSCFG0			
0x01C1 5000	0x01C1 FFFF						
0x01C2 0000	0x01C2 0FFF	4K		Timer0			
0x01C2 1000	0x01C2 1FFF	4K		Timer1			
0x01C2 2000	0x01C2 2FFF	4K		I2C 0			
0x01C2 3000	0x01C2 3FFF	4K		RTC			
0x01C2 4000	0x01C3 FFFF						
0x01C4 0000	0x01C4 0FFF	4K		MMC/SD 0			
0x01C4 1000	0x01C4 1FFF	4K		SPI 0			
0x01C4 2000	0x01C4 2FFF	4K		UART 0			
0x01C4 3000	0x01CF FFFF						
0x01D0 0000	0x01D0 0FFF	4K		McASP 0 Control			
0x01D0 1000	0x01D0 1FFF	4K		McASP 0 AFIFO Ctrl			

(1) The DSP L2 ROM is used for boot purposes and cannot be programmed with application code

TMS320C6748 Fixed- and Floating-Point DSP
 (SPRS590F), Texas Instruments,
 2009–REVISED MARCH 2014
 (<http://www.ti.com>)



Структура памяти

Память TMS320C6713 строится по **двухуровневой архитектуре**.

Память **первого уровня L1** относительно небольшая по объему, но с высоким быстродействием (чтение/запись за 1 такт). Она используется как **кэш**. Это означает, что в нее копируется фрагмент данных (команд), используемый в текущих вычислениях, для быстрого доступа.

Память **второго уровня L2** имеет относительно большой объем, но низкую скорость доступа. Для работы с этой памятью на скорости ядра необходима промежуточная **кэш-память (L1)**.

Часть памяти уровня L2 может конфигурироваться как **кэш** для работы с еще более медленной **внешней памятью**.

Для программиста **кэш-память «невидима»**. Он просто работает с внутренней памятью, но должен помнить о наличии механизма кэширования.



Система команд ЦСТ TMS320C67x

Синтаксис команды:

Мнемоника	Исполняющий вычислительный блок	Операнды источники	Регистр назначения
-----------	---------------------------------------	-----------------------	-----------------------

Пример:

LDW .D1T1 *A1,A2

LDDW .D2T1 *B1,A0

STW .D2T2 A0,*B7

STDW .D1T2 B0,*A7



Система команд ЦСТ TMS320C67x

Команды, выполняемые блоком .D

Table C-1. Instructions Executing in the .D Functional Unit

Instruction	Format	Instruction	Format
ADD	Figure C-1, Figure C-2	MV	Figure C-1, Figure C-2
ADDAB	Figure C-1, Figure C-3	MVK	Figure C-1
ADDAD	Figure C-1	NOT	Figure C-2
ADDAH	Figure C-1, Figure C-3	OR	Figure C-2
ADDAW	Figure C-1, Figure C-3	STB	Figure C-4
ADD2	Figure C-2	STB (15-bit offset) ⁽¹⁾	Figure C-5
AND	Figure C-2	STDW	Figure C-6
ANDN	Figure C-2	STH	Figure C-4
LDB	Figure C-4	STH (15-bit offset) ⁽¹⁾	Figure C-5
LDB (15-bit offset) ⁽¹⁾	Figure C-5	STNDW	Figure C-7
LDBU	Figure C-4	STNW	Figure C-4
LDBU (15-bit offset) ⁽¹⁾	Figure C-5	STW	Figure C-4
LDDW	Figure C-6	STW (15-bit offset) ⁽¹⁾	Figure C-5
LDH	Figure C-4	SUB	Figure C-1, Figure C-2
LDH (15-bit offset) ⁽¹⁾	Figure C-5	SUBAB	Figure C-1
LDHU	Figure C-4	SUBAH	Figure C-1
LDHU (15-bit offset) ⁽¹⁾	Figure C-5	SUBAW	Figure C-1
LDNDW	Figure C-7	SUB2	Figure C-2
LDNW	Figure C-4	XOR	Figure C-2
LDW	Figure C-4	ZERO	Figure C-1, Figure C-2
LDW (15-bit offset) ⁽¹⁾	Figure C-5		

⁽¹⁾ D2 only

TMS320C674x DSP
CPU and Instruction Set
Reference Guide,
Texas Instruments,
Literature Number: SPRUFE8B
July 2010



Система команд ЦСТ TMS320C67x

Режимы адресации

- непосредственная;
- регистровая;
- прямая;
- косвенная.

Режимы косвенной адресации

- косвенная адресация;
- косвенная адресация со сдвигом адреса;
- косвенная адресация с пред-/пост- модификацией адреса.



Система команд ЦСТ TMS320C67x

При объединении команд в пакеты выполнения следует быть внимательным к совместному использованию ресурсов:

```
LDW      .D2T2      *B3++[1],B5
| | LDW      .D1T1      *A2,B2
| | MPYSP     .M1       A1,A1,A2
| | STW      .D2T2      B2,*B7++[1]
```

```
LDW      .D2T1      *B3++[1],A5
| | LDW      .D1T1      *A2,B2
| | MV       .L2X      A2,B2
| | ADD      .S2       B2,1,B2
```



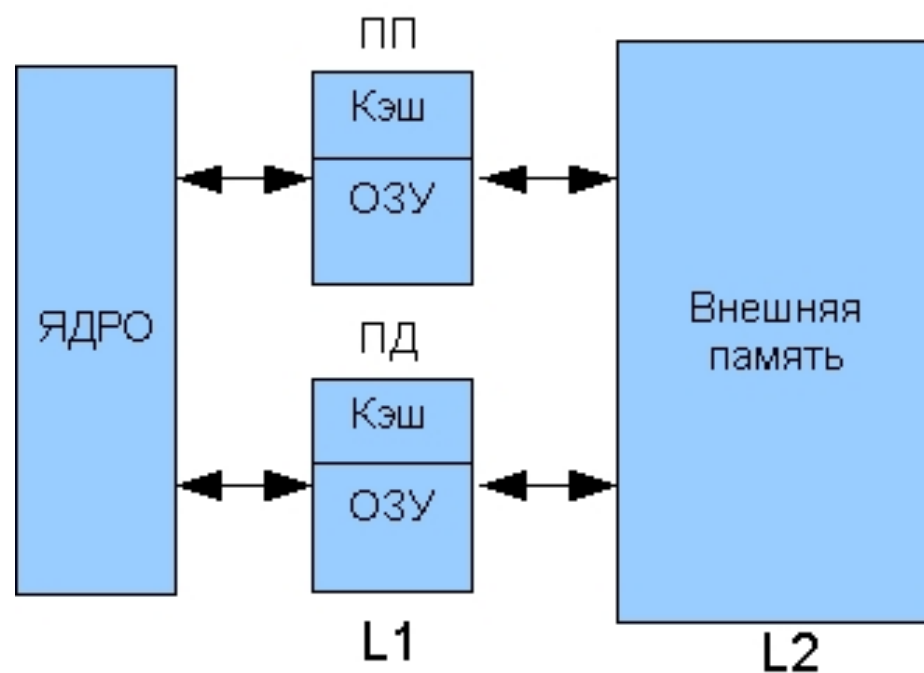
Система команд ЦСП TMS320C67x

Примеры !



Система команд ЦСТ TMS320C67x

Иерархическая архитектура памяти и кэш-память





Система команд ЦСП TMS320C67x

Иерархическая архитектура памяти и кэш-память

Кэширование программ и данных имеет основной целью ускорение работы быстрого ядра с медленной памятью большого объема

Эффективность применения кэш основана на двух фактах - локализованности программ/данных во времени и в пространстве (памяти).



Система команд ЦСТ TMS320C67x

Иерархическая архитектура памяти и кэш-память

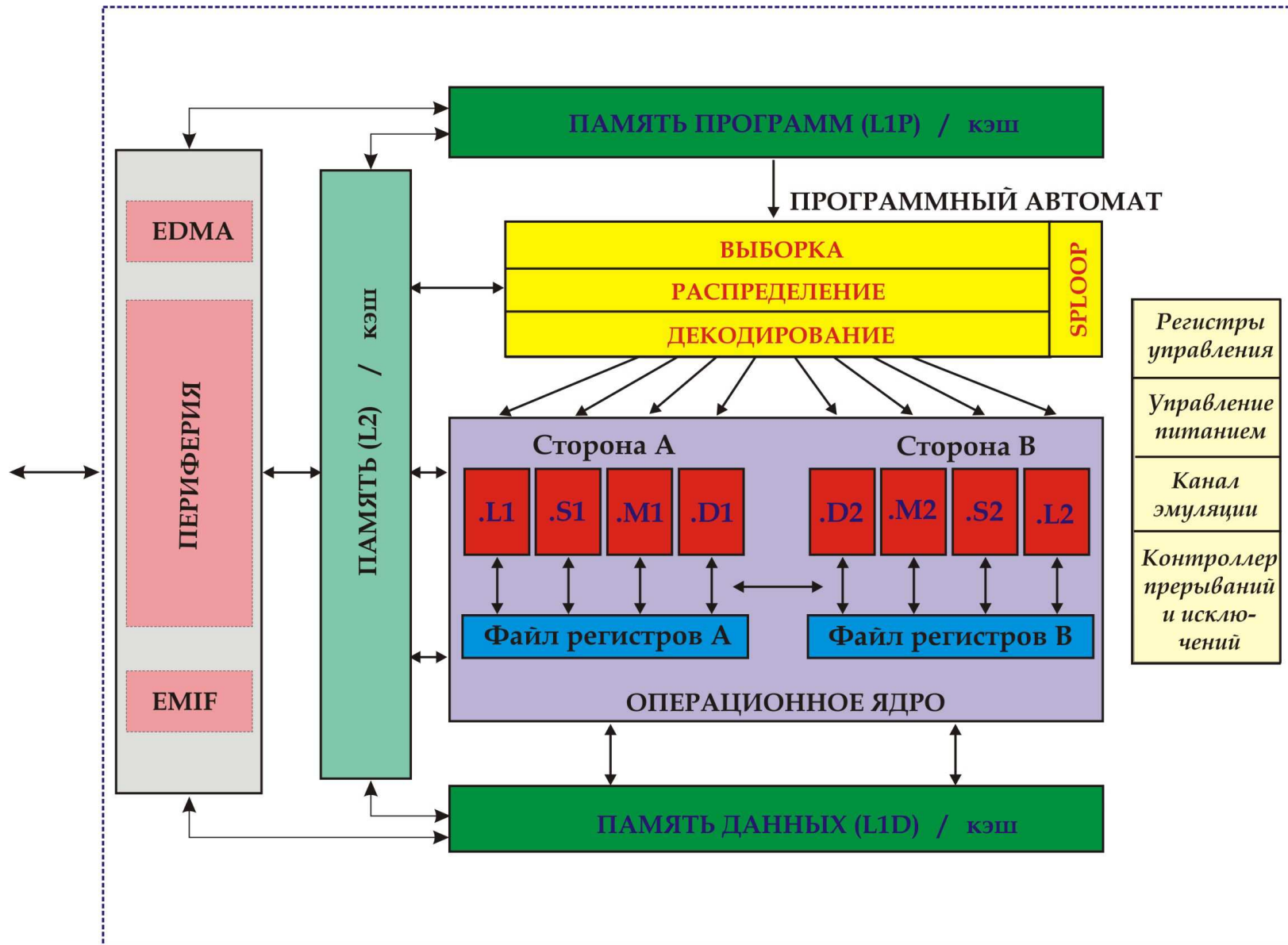
Кэш с прямым отображением в памяти (direct-mapped cache)

Ассоциативная кэш-память, 2-вариантная / 4-вариантная (2-way / 4-way set-associative cache)

	way1	way2
set1	line0	line0
set2	line1	line1
set3	line2	line2

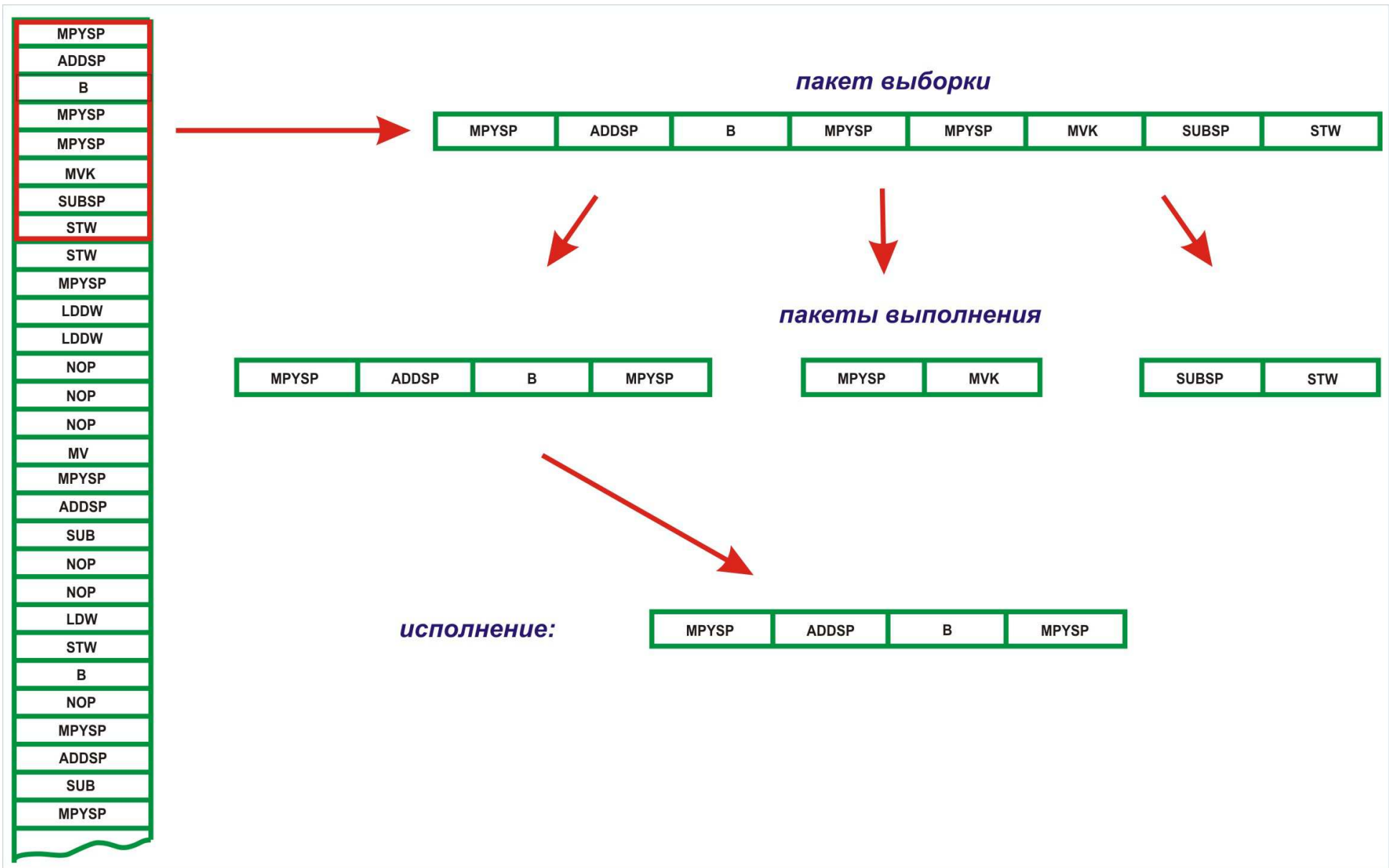


Архитектура ЦСТ TMS320C67x: программный автомат





Архитектура ЦСТ TMS320C67x: программный автомат





Программный автомат

Программа, предназначенная для выполнения на ЦСП, располагается в памяти процессора и представляет собой последовательность команд.

Выборку команд из памяти и организацию работы функциональных узлов в соответствии с выбранной командой берет на себя программный автомат (секвенсор команд).

Выборка команд из памяти осуществляется всегда по восемь последовательно расположенных инструкций. Такая группа команд называется пакетом выборки.

В лучшем случае все восемь команд могут быть выполнены одновременно на восьми вычислительных блоках. Тогда весь пакет выборки поступает на выполнение и называется пакетом выполнения.



Если все команды пакета выборки не могут выполняться параллельно, то пакет разбивается на группы команд, которые могут выполняться одновременно, и каждая из этих групп образует пакет выполнения.

Например, если в пакете выборки оказалось три команды умножения, то, очевидно, что его можно обработать только в два этапа, поскольку процессор имеет два блока умножения.

Ограничить возможности параллельного выполнения выбранных команд может и какой-либо другой ресурс, используемый совместно несколькими командами.

Выборка следующего пакета команд из памяти процессора производится только после того, как выполняться все команды текущего пакета выборки.



Архитектура ЦСП TMS320C67x: программный автомат





Процесс обработки команд в ЦСП TMS320C67x разбит на 3 этапа:

- **выборка**
- **декодирование**
- **исполнение**

На каждом этапе команды проходят несколько фаз:

- выборка - 4 фазы;
- декодирование - 2 фазы;
- исполнение - от 1 до 10 фаз! (в зависимости от типа команды)

Замечание. В этих и нижеследующих рассуждениях под командой следует понимать пакет выполнения, так как команды одного пакета выполнения обрабатываются одновременно и независимо.



Время прохождения командой одной фазы обработки является основной единицей измерения временных интервалов для ЦСП и называется **командным циклом процессора**. Оно фиксировано и напрямую определяется **тактовой частотой процессора** - одной из основных характеристик быстродействия. Часто командный цикл процессора является **равным такту процессора** - величине, обратной тактовой частоте.

Все команды проходят первые 7 фаз. Некоторые команды проходят большее число фаз.

Рассмотрим те команды, которые проходят только 7 фаз. Каждая из них выполняется за 7 циклов. Однако следующие друг за другом, скажем 10 команд, выполняются не за 70, а за 10 циклов.

Это оказывается возможным за счет применения принципа **конвейерной обработки команд**: если i -я команда прошла j -ю фазу обработки и поступила на $(j+1)$ -ю фазу, то $(i+1)$ -я команда может поступать на j -ю фазу обработки, не дожидаясь полного выполнения i -ой команды.



С учетом принципа конвейерной обработки, команды, полностью выполняемые за 7 циклов (тактов), называются **однотактными** (1 команды выполняется за 7 циклов; 2 команды - за 8 циклов; 10 команд - за 10 циклов; 11 команд - за 11 циклов и так далее).

Однако не все команды полностью выполняются за 7 циклов. Возникает необходимость классифицировать команды (**НЕ пакеты выполнения!**) по различным типам.

Принадлежность команды к тому или иному типу определяет основные параметры выполнения команды, к которым относятся:

- время задержки результата
- время использования вычислительного блока



Время задержки результата - это число командных циклов процессора, дополнительно требуемых для получения результата выполнения команды.

Время использования вычислительного блока - это число командных циклов процессора, в течение которых данная команда занимает соответствующий ей вычислительный блок.

Перечислим **основные типы команд**:



Архитектура ЦСП TMS320C67x: программный автомат

Single Cycle	команды, выполняемые за один такт. Время задержки результата равно 0. Время использования вычислительного блока – 1 такт.
Store	команды записи содержимого регистров в память данных процессора. Время задержки результата равно 0. Время использования вычислительного блока 1 такт.
Load	команды чтения данных из памяти процессора в регистры операционного ядра. Данный тип команд характеризуется временем задержки результата в 4 такта и временем использования вычислительного блока – 1 такт.
Branch	команды перехода (ветвления) с текущей строки программы к строке указанной в команде. Команды ветвления требуют 5 тактов задержки результата выполнения. Время использования вычислительного блока равно 1.
4-cycle	команды, выполняемые за 4 такта – к ним, в первую очередь, относятся команды сложения и умножения чисел в формате с плавающей точкой. Время задержки результата составляет 3 такта, а время использования блока – 1 такт.