

## VIII. Организация памяти ЭВМ

Памятью ЭВМ называется совокупность устройств, служащих для запоминания, хранения и выдачи информации.

Характеристики памяти ЭВМ:

-Назначение.

-Информационная емкость.

-Информационная емкость читаемого слова.

-Способ доступа.

-Быстродействие.

-Физический способ хранения информации.

# Классификация запоминающих устройств по способу доступа.

## - Адресные ЗУ

Постоянные ЗУ, ПЗУ (ROM)

ЗУ с произвольным доступом (RAM)

## - Ассоциативные ЗУ

Полностью ассоциативные ЗУ

Ассоциативные ЗУ с прямым размещением

Наборно-ассоциативные ЗУ

## - Последовательные ЗУ

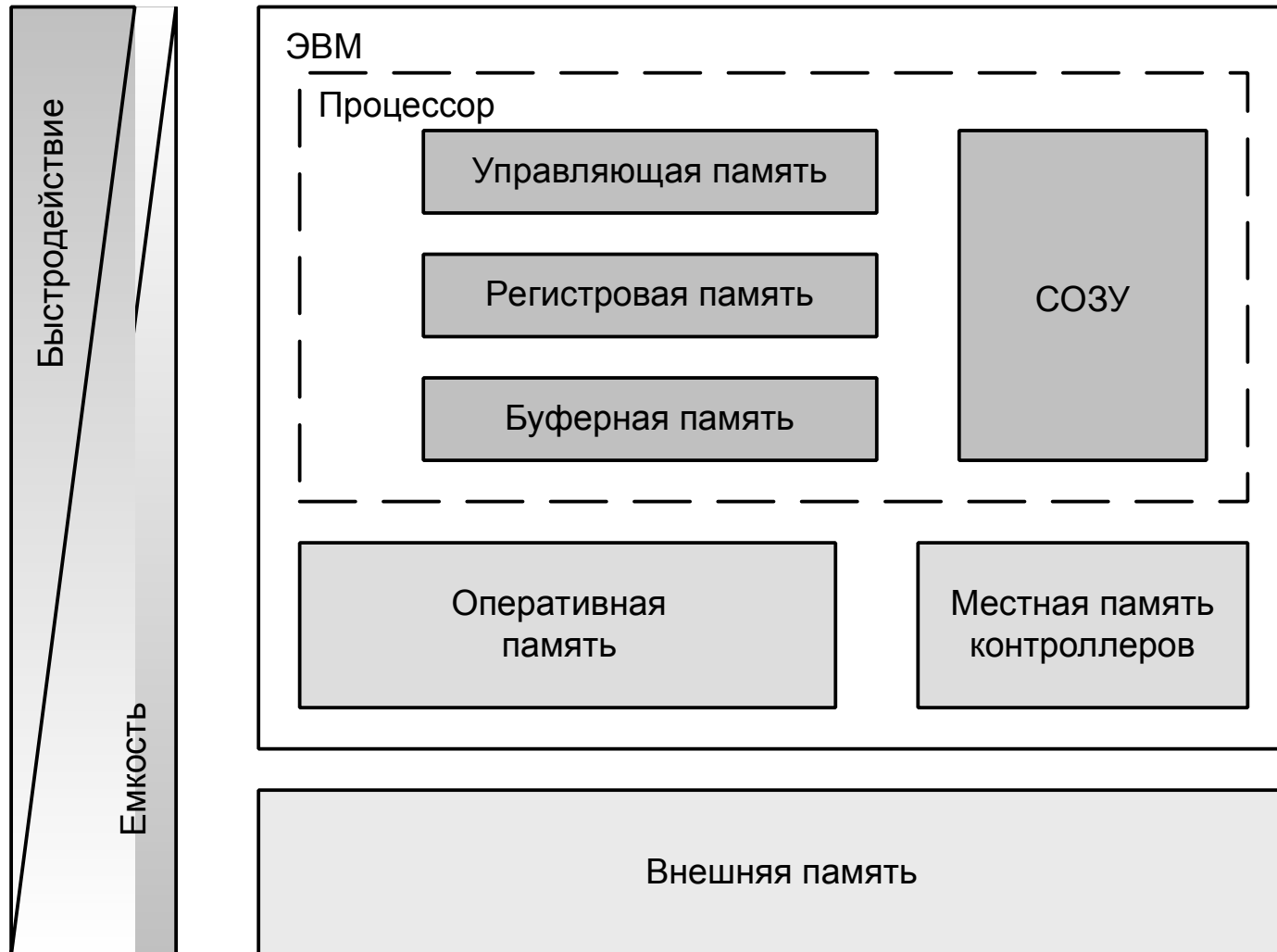
FIFO

LIFO

Файловые

Циклические

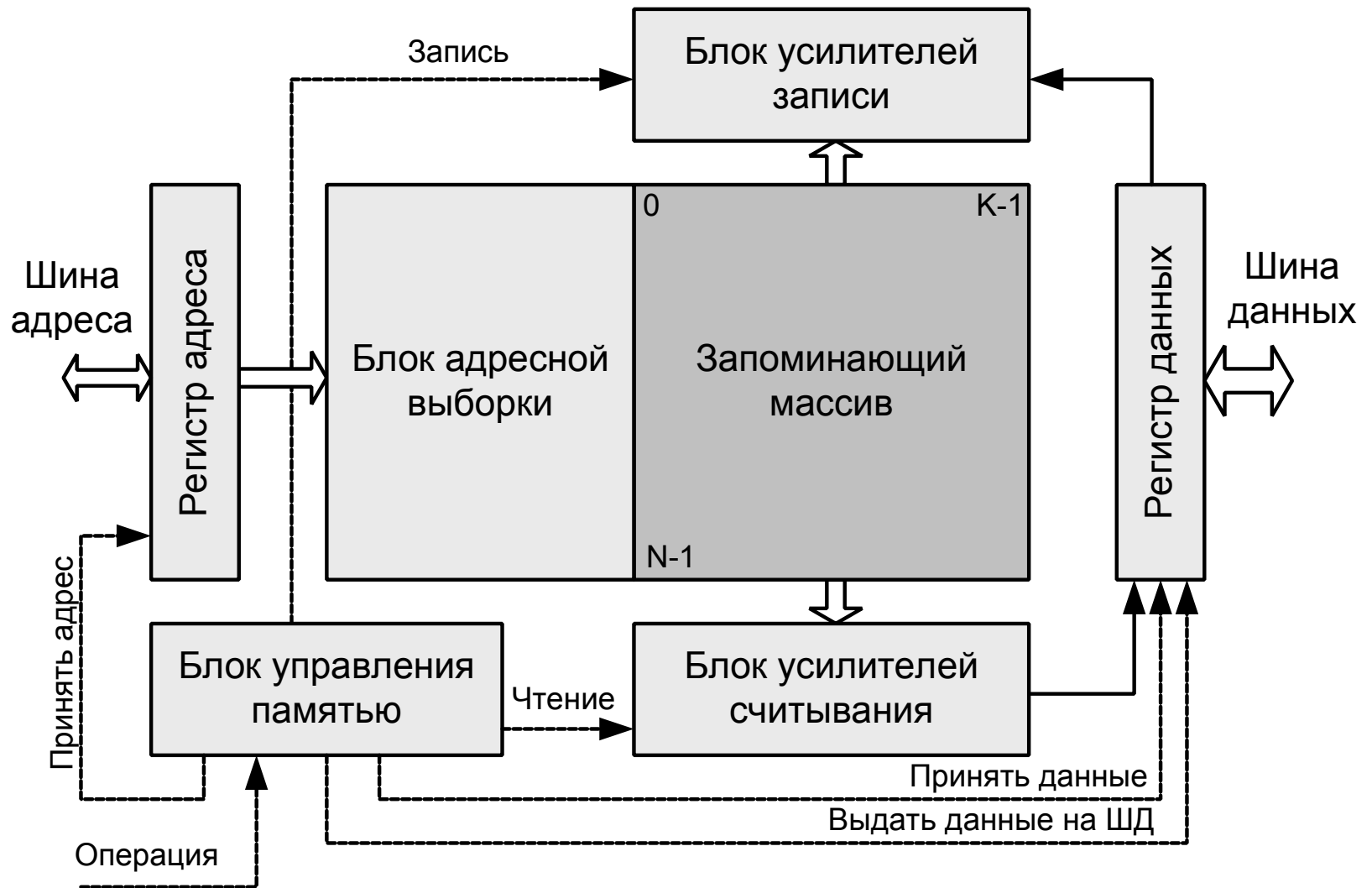
# Классификация запоминающих устройств по назначению.



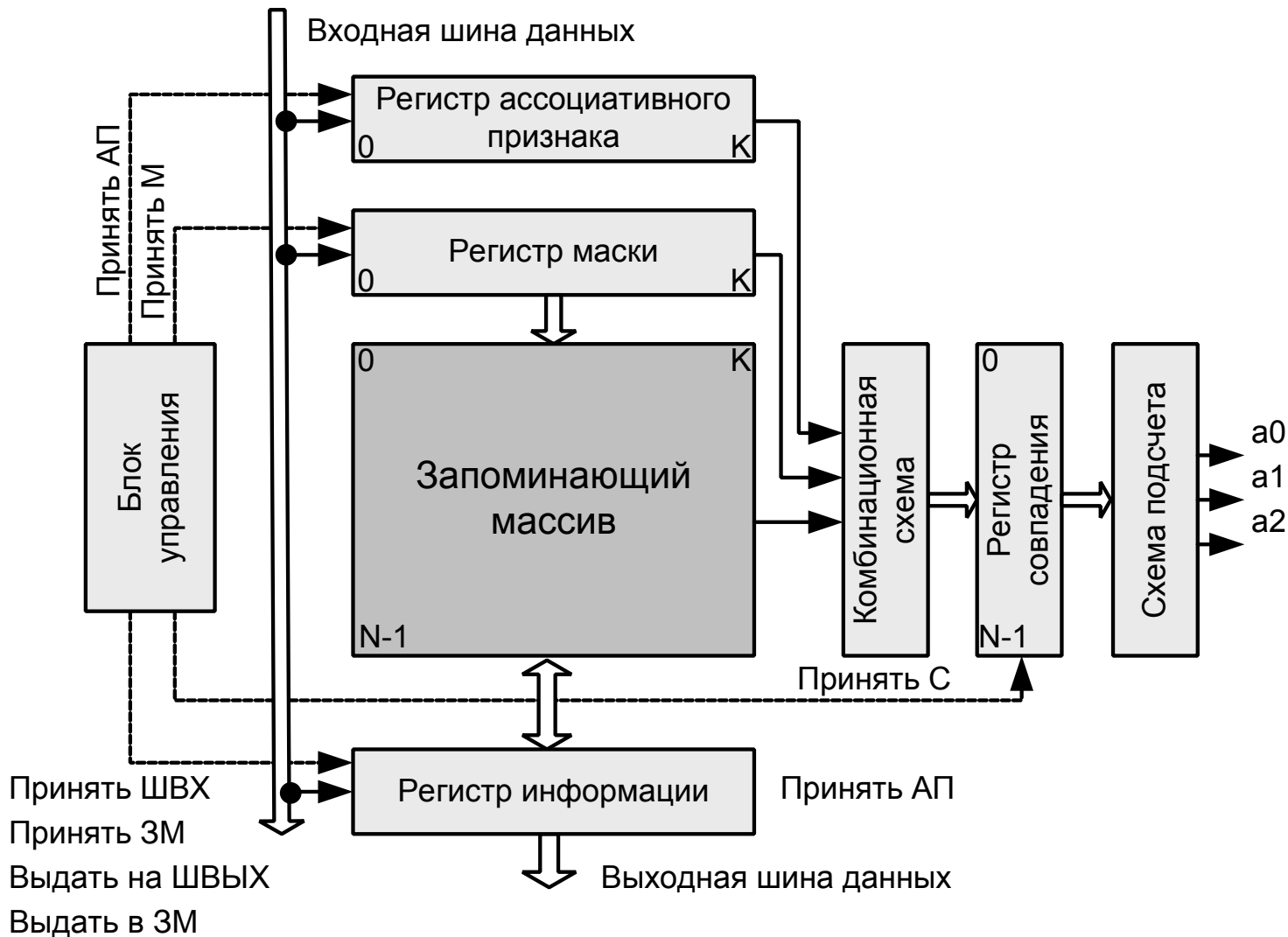
## Латентность при обращении к подсистеме памяти

Тип обращения к памяти	Объем памяти, байт	Латентность, такты процессора
Регистры процессора	$2^6 \dots 2^{10}$	1
Кэш первого уровня	$2^{14} \dots 2^{16}$	2..4
Кэш второго уровня	$2^{15} \dots 2^{22}$	10..12
Кэш третьего уровня	$2^{21} \dots 2^{26}$	15..50
ОЗУ на одном кристалле с процессором при попадании в TLB (доступ по случайным адресам)	$2^{14} \dots 2^{24}$	10..75
Внешнее ОЗУ при попадании в TLB (доступ по случайным адресам)	$2^{20} \dots 2^{40}$	200..400
Внешнее ОЗУ при промахе в TLB (доступ по случайным адресам)	$2^{20} \dots 2^{40}$	2000..2500
Внешнее ОЗУ при выгруженной во внешний Flash диск странице	$2^{30} \dots 2^{42}$	$1 \cdot 10^5 \dots 1 \cdot 10^6$
Внешнее ОЗУ при выгруженной во внешний жесткий диск странице	$2^{30} \dots 2^{50}$	$1 \cdot 10^6 \dots 1 \cdot 10^8$

# Обобщенная схема адресного ЗУ

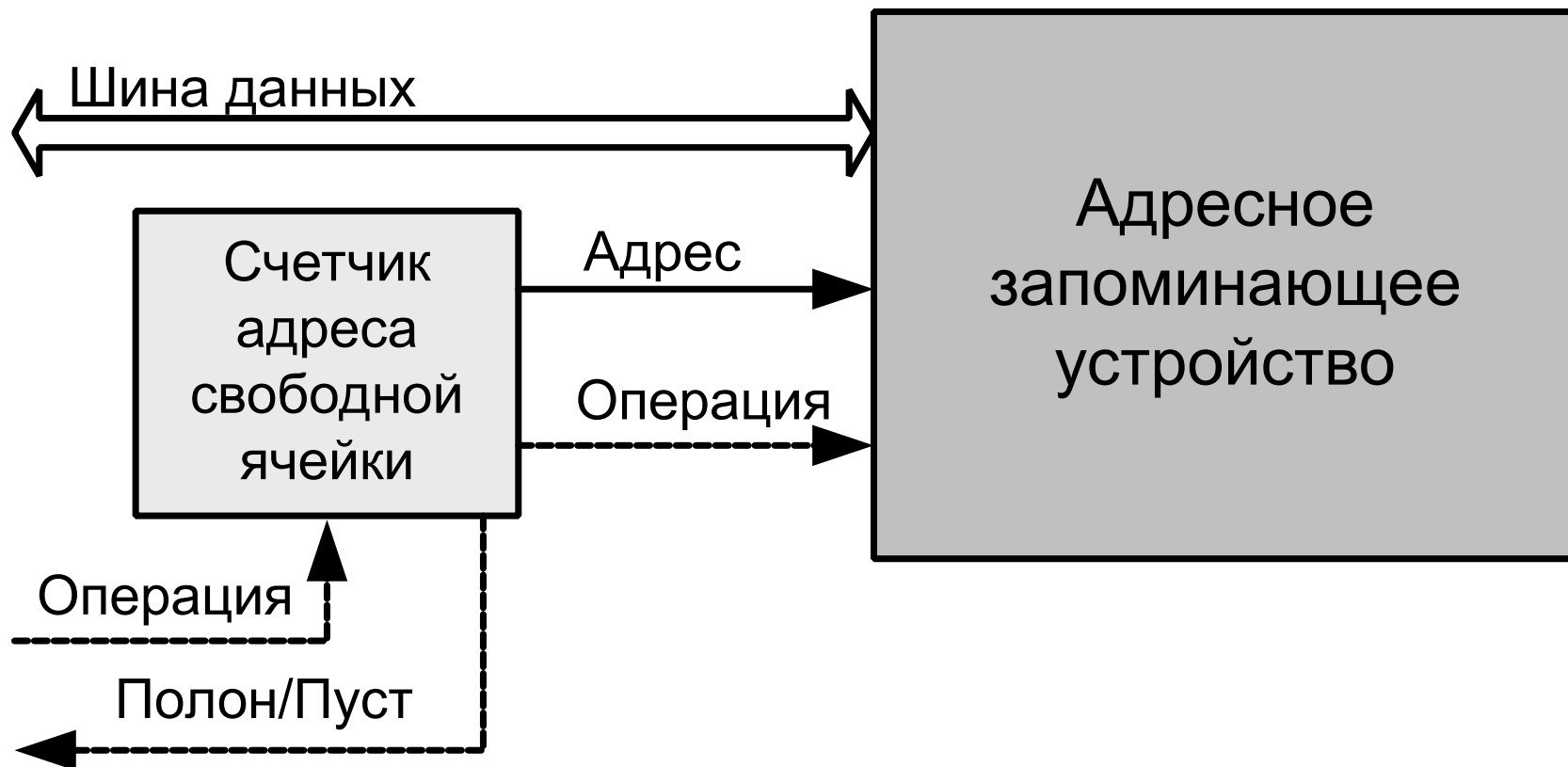


# Обобщенная схема ассоциативного ЗУ

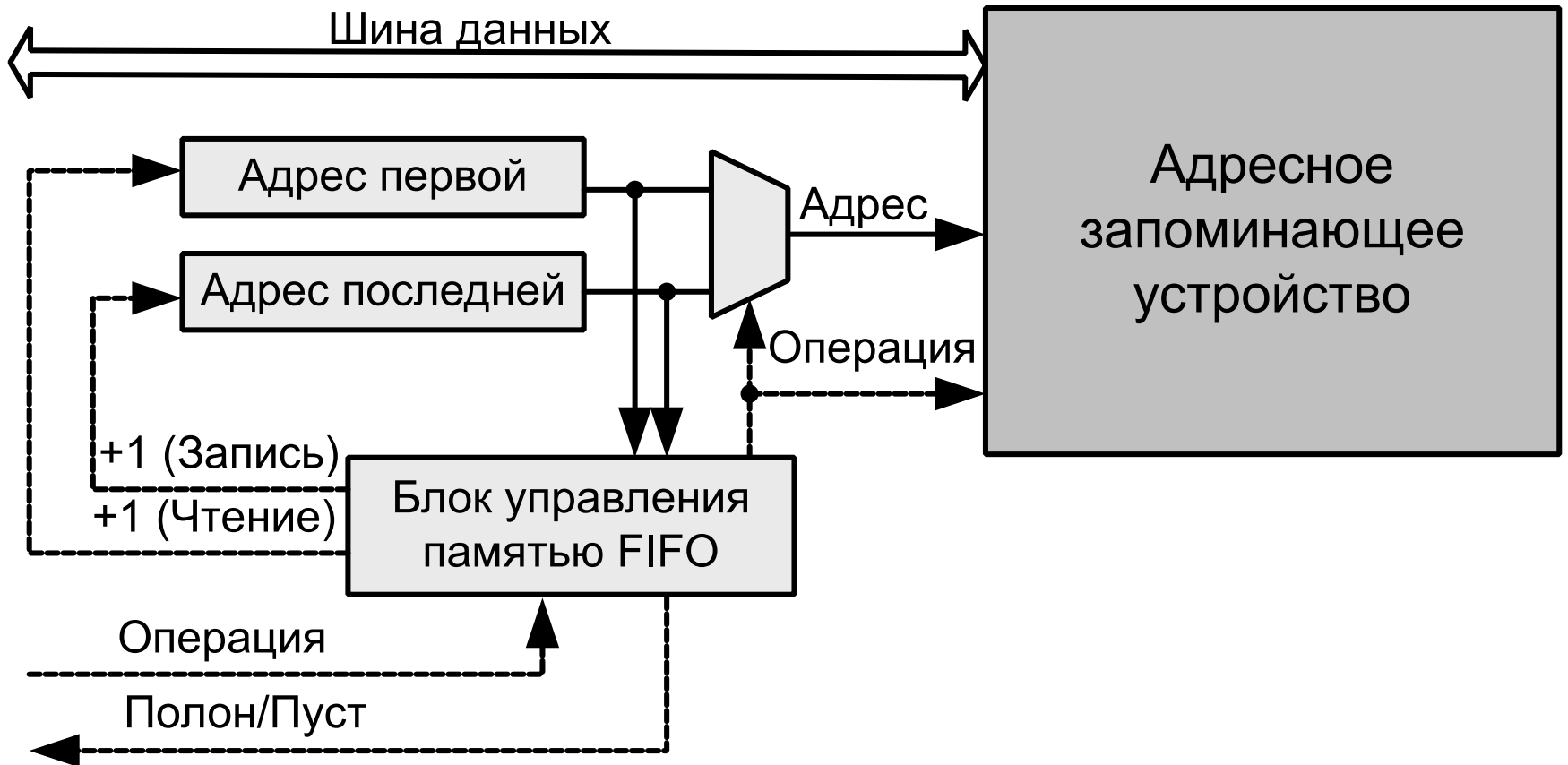


# Обобщенная схема последовательного ЗУ

Стек (память типа LIFO)



# Буфер (память типа FIFO)





# Адресные запоминающие устройства

## Постоянные ЗУ, ПЗУ (ROM)

МПЗУ (MROM)

ППЗУ (PROM)

РПЗУ-УФ (EPROM)

ОПРПЗУ-УФ (EPROM-OTP)

РПЗУ-ЭС (EEPROM)

FLASH

## ЗУ с произвольным доступом (RAM)

### Динамические ЗУПД (DRAM)

Использующие кучность  
адресов

FPM DRAM

EDO DRAM

BEDO DRAM

SDRAM

DDR SDRAM

RDRAM

Не использующие кучность  
адресов

DRAM

RLDRAM

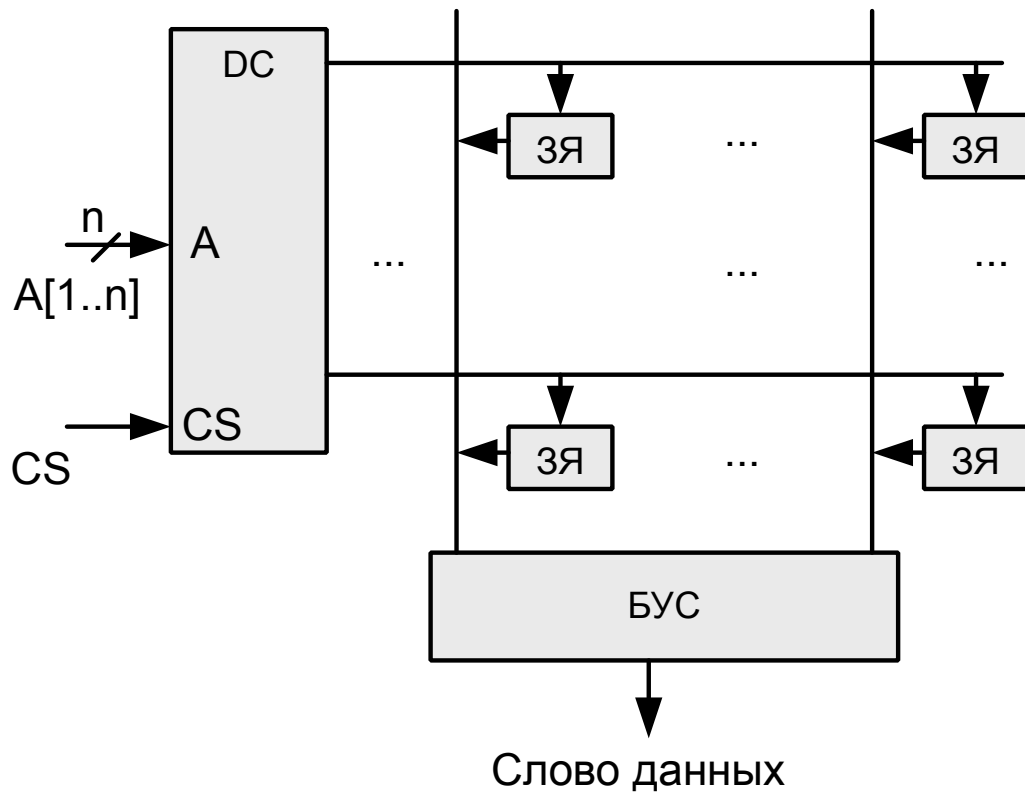
### Статические ЗУПД (SRAM)

Асинхронные

Синхронные

# Организация запоминающих массивов адресных ЗУ

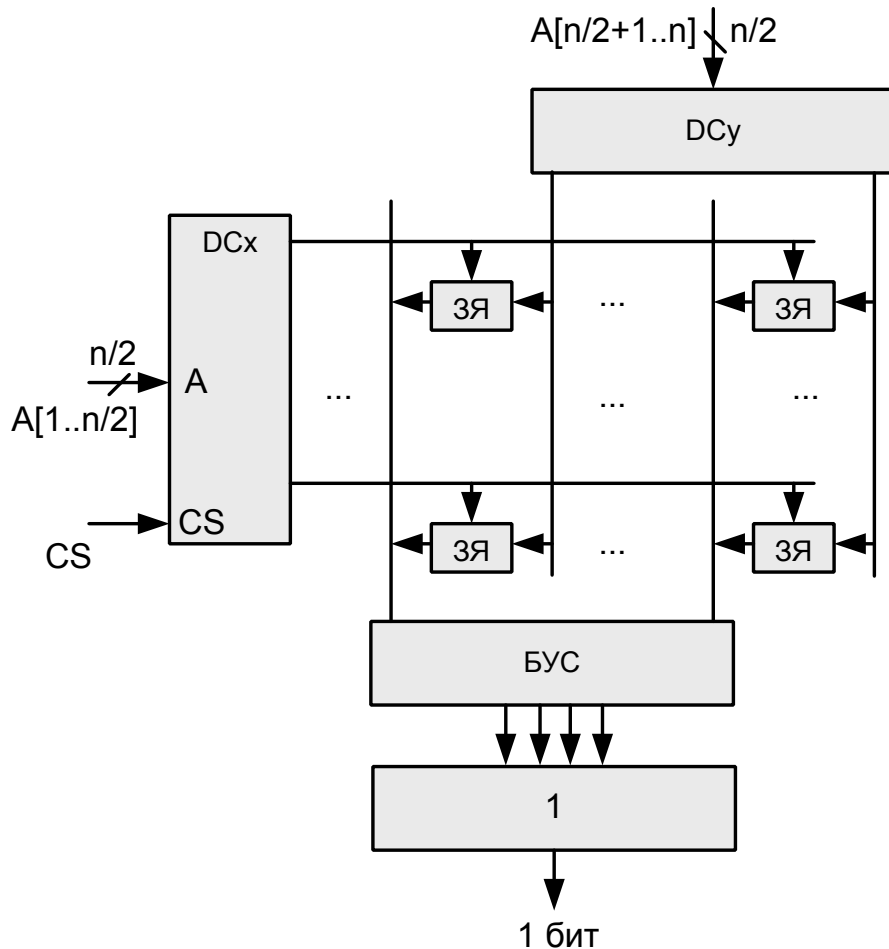
## Структура ЗМ типа 2D



Количество выходов дешифратора равно количеству слов в памяти ( $2^n$ )

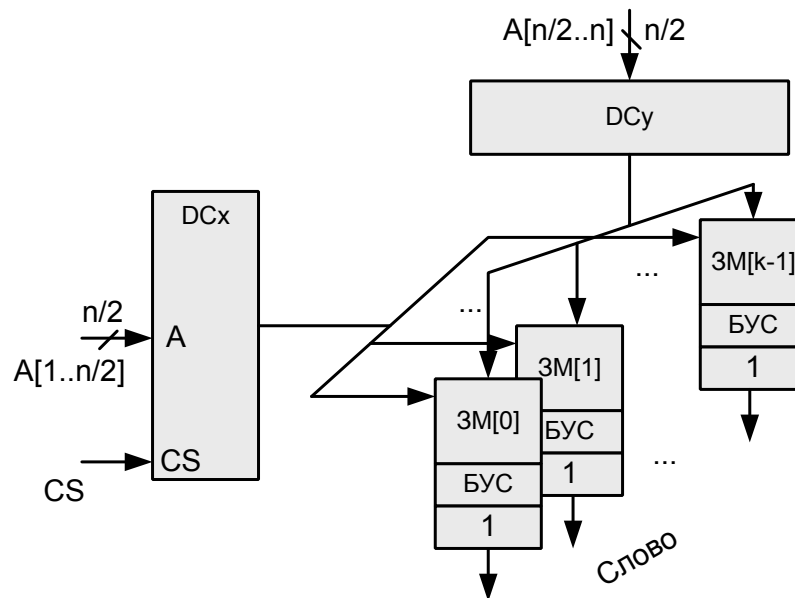
Структура применима только для малоразмерных ЗУ

# Структура 3М типа 3D

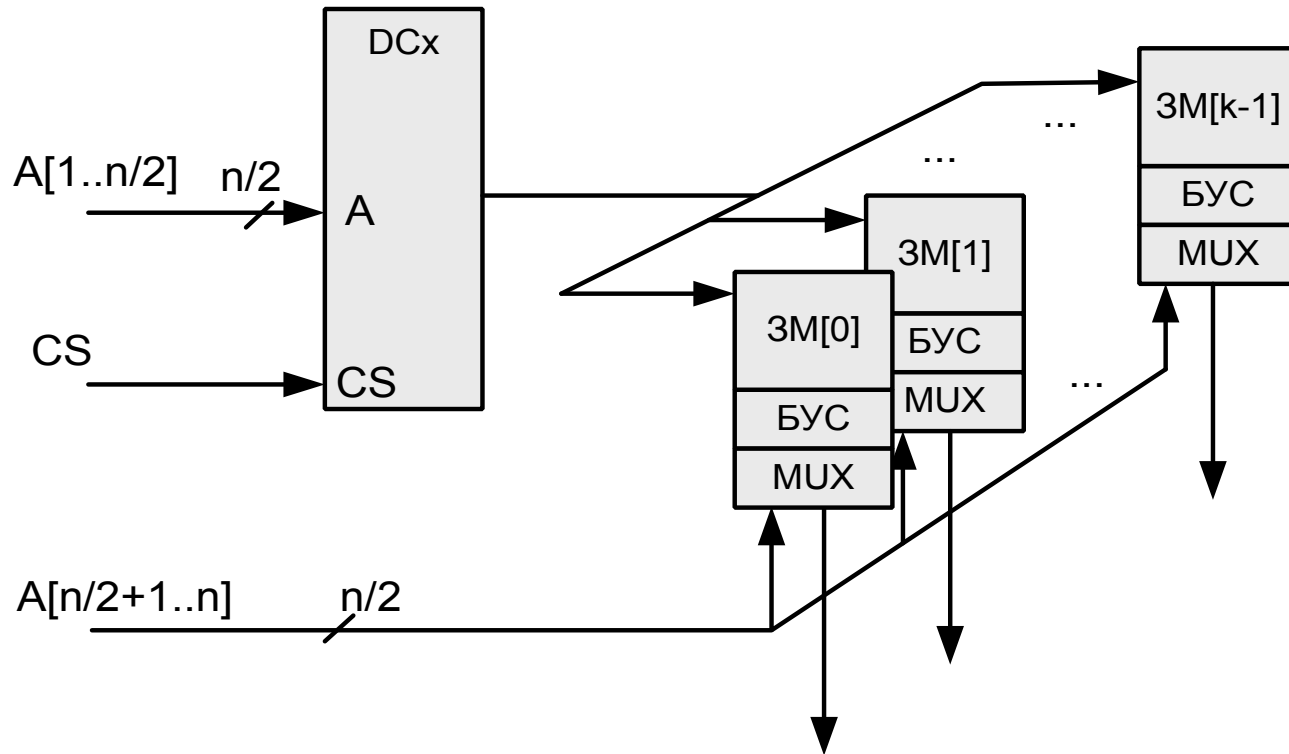


Адрес делится на две части (двухкоординатная выборка).

Количество выходов дешифраторов:  $2^{n/2} + 2^{n/2}$



## Структура 3М типа 2DM

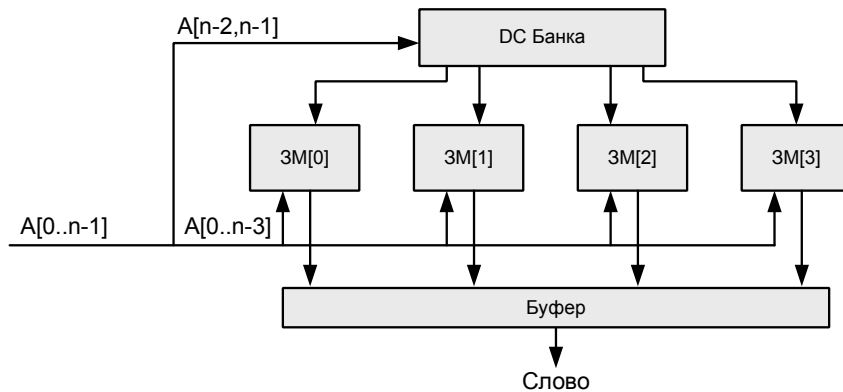


Мультиплексоры позволяют выбрать один из  $2^{n/2}$  разрядов каждом из запоминающих массивов

- Размеры массивов близки к оптимальным.
- Количество линий записи/считывания минимально.

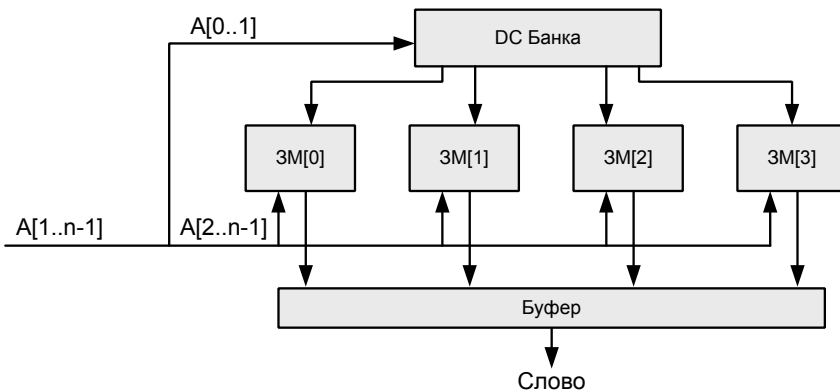
# Расслоение памяти

## Блочное разделение адреса



Номер банка определяется старшей частью адреса.

## Циклическое разделение адреса



Номер банка определяется младшей частью адреса

# Блочное-циклическое разделение адреса

Банк [0]		Банк [1]	
ЗМ [0]	ЗМ [1]	ЗМ [0]	ЗМ [1]
0	1	2	3
4	5	6	7
...	...	...	...

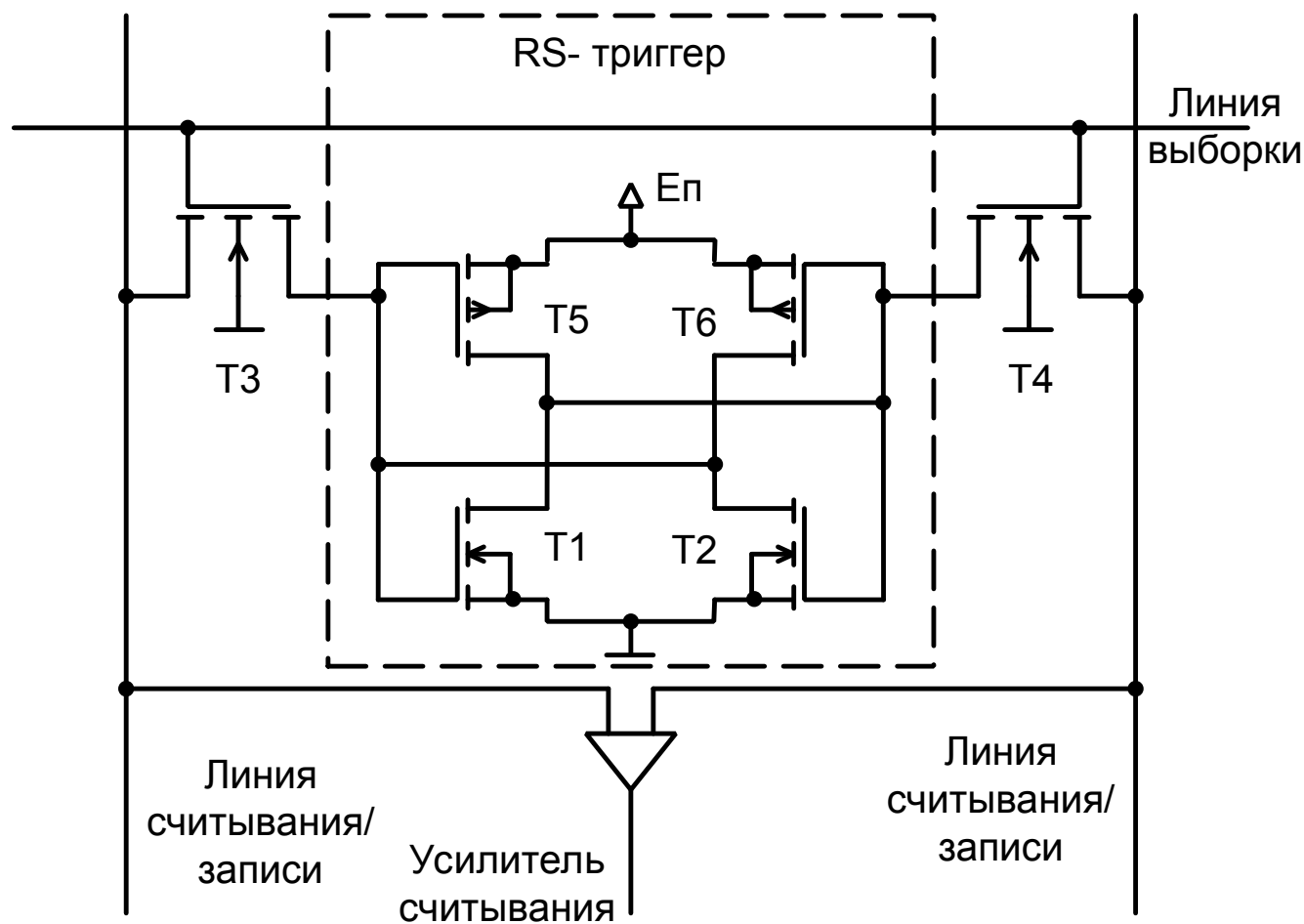
Блочное-циклический способ обеспечивает возможность пакетной передачи и ускоряет доступ при кучности адресов

## Пример разделения адреса в SDRAM (P111, P4)

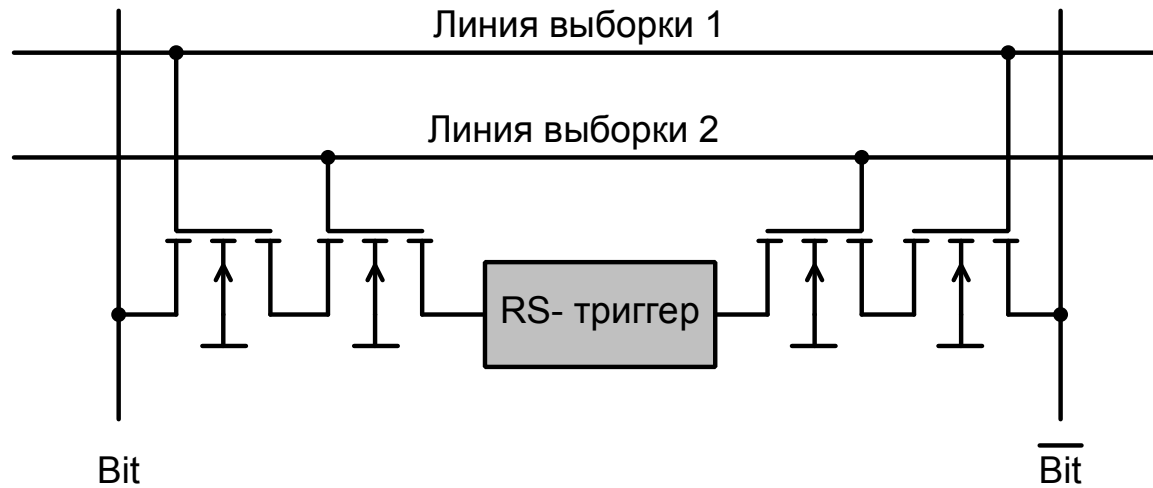


# Статические ЗУ с произвольной выборкой (SRAM)

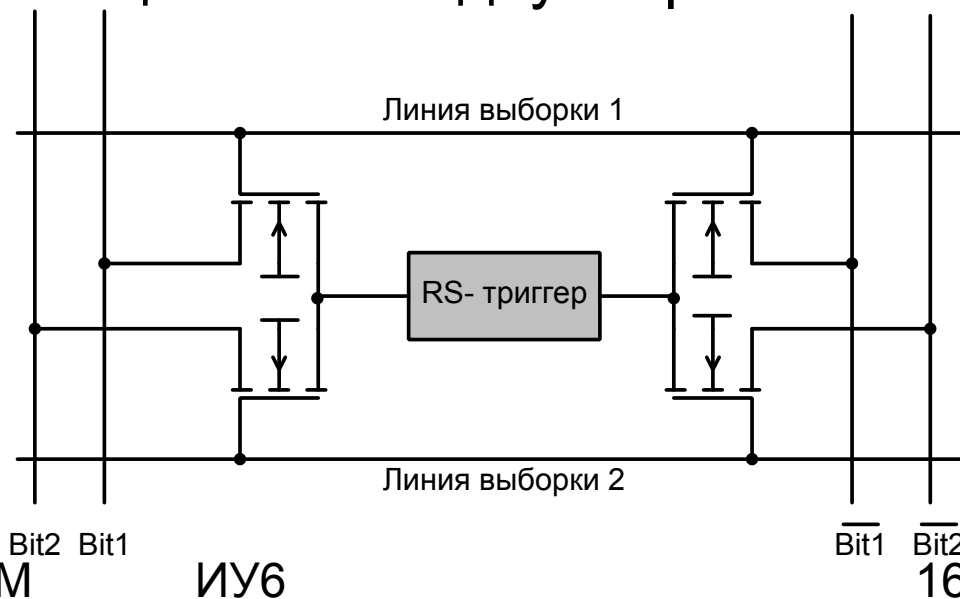
## Запоминающая ячейка статической памяти



# Запоминающая ячейка с двухкоординатной выборкой

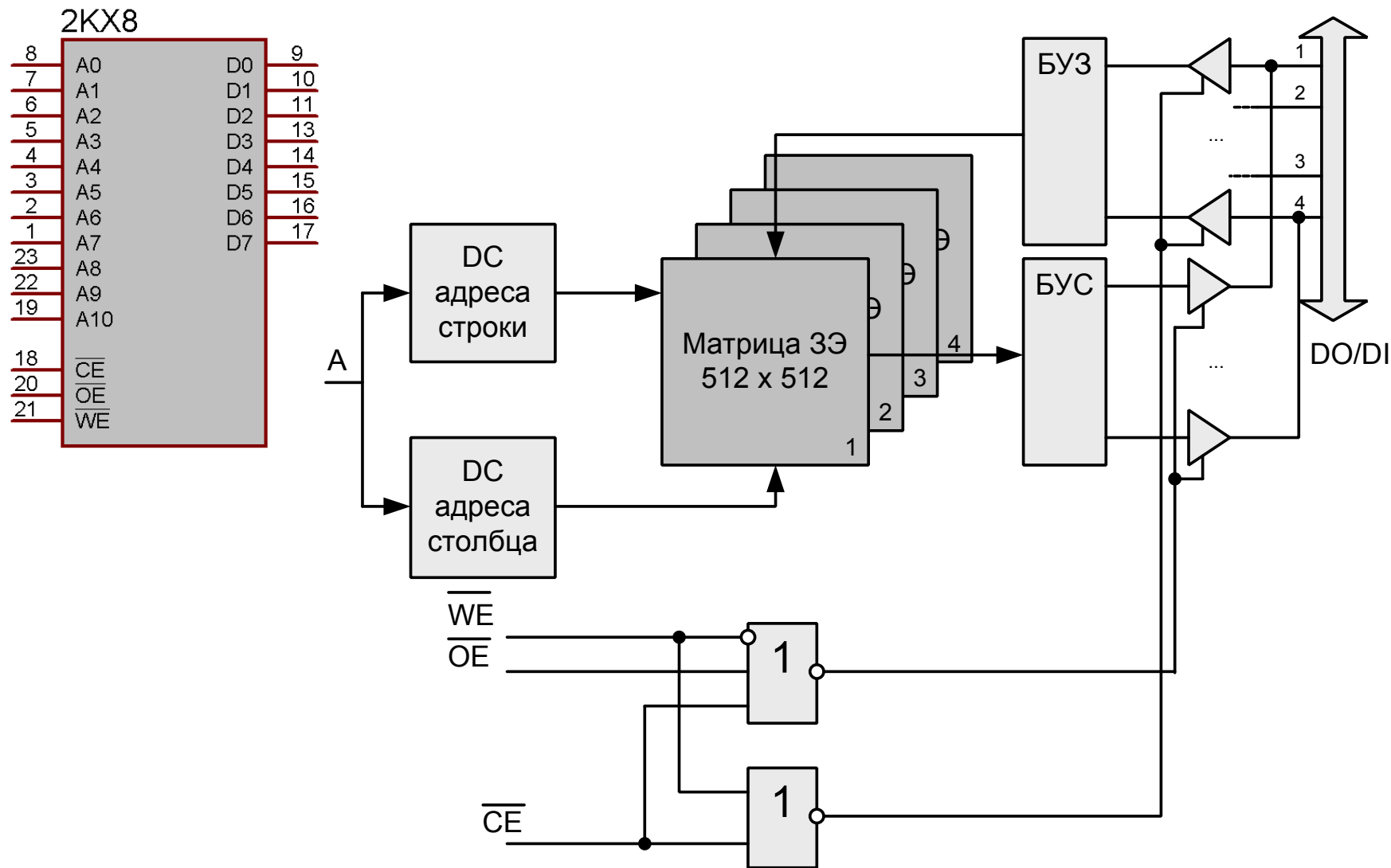


# Запоминающая ячейка двухпортовой выборкой

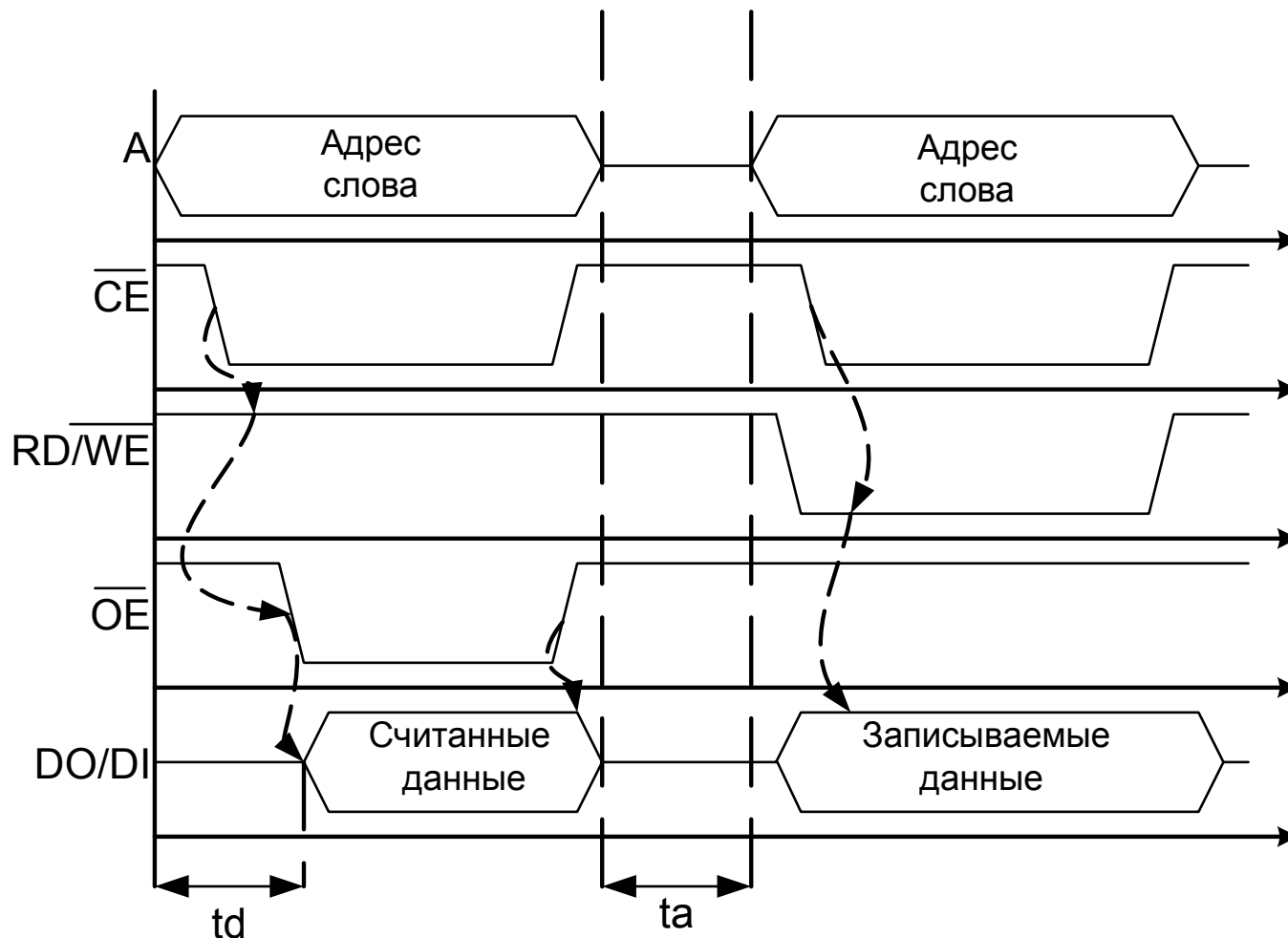
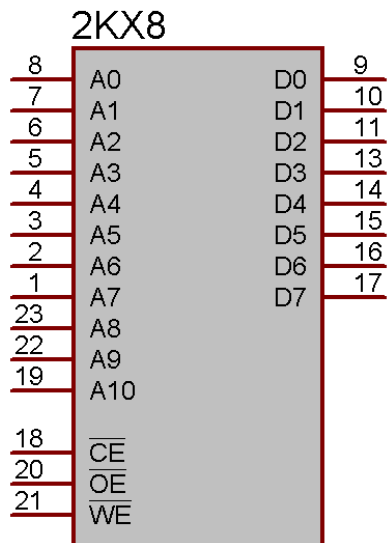




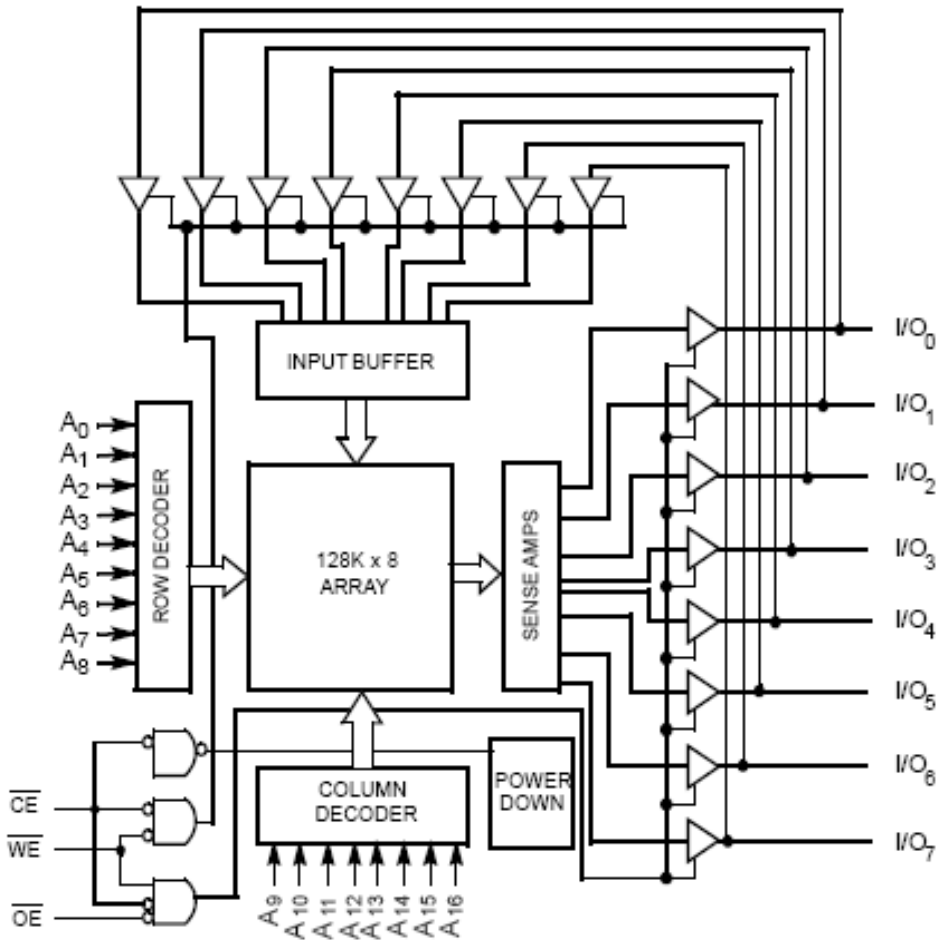
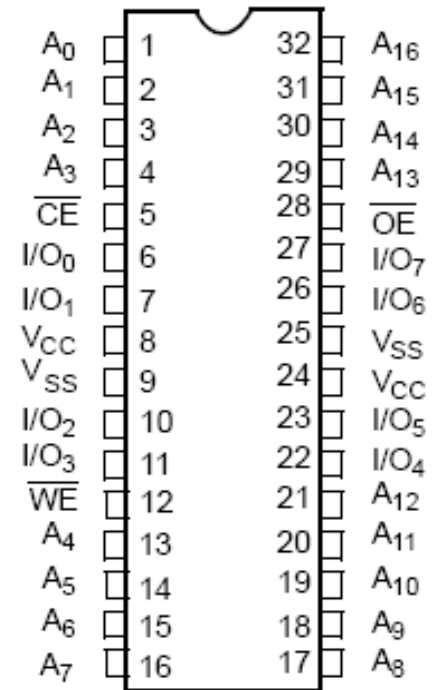
# Микросхема статической памяти



# Диаграмма работы статической памяти



SOJ/TSOP II  
Top View

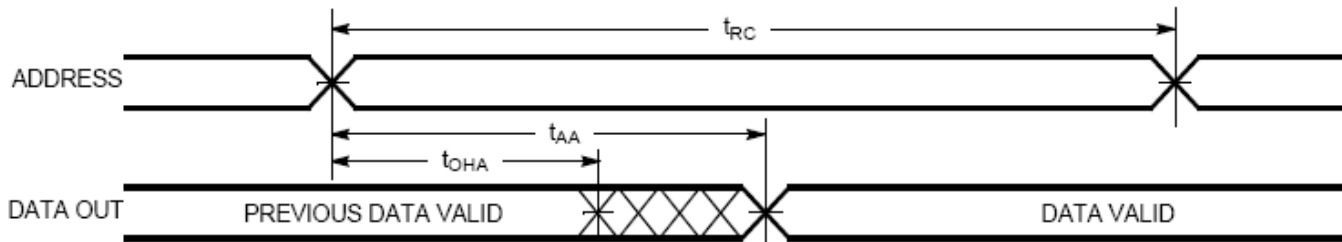


ПРИМЕР

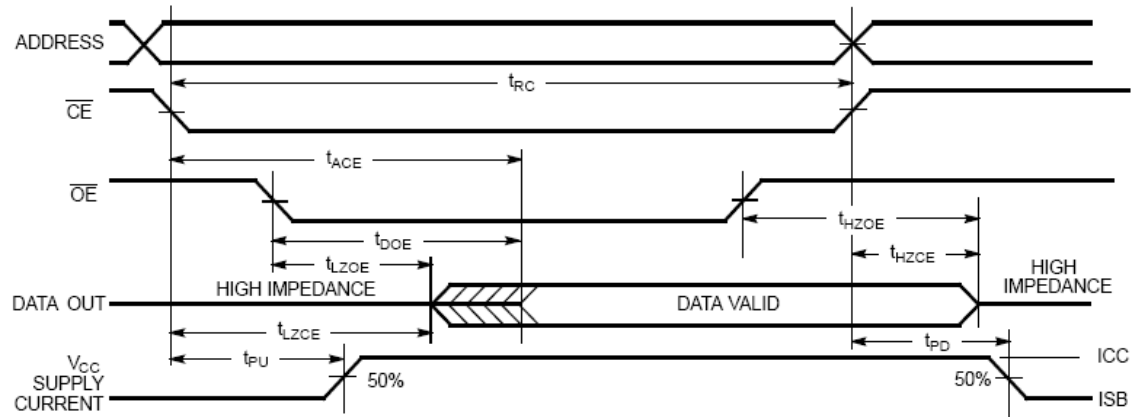
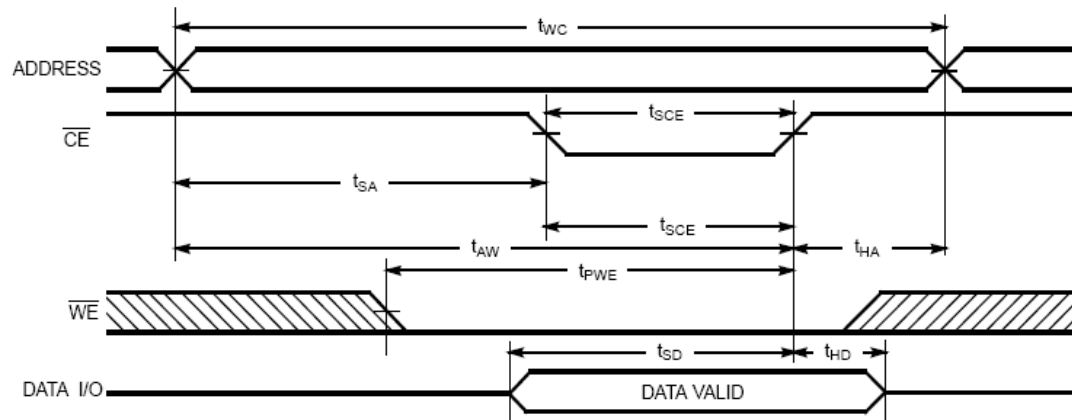
**Truth Table**

$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{WE}}$	I/O <sub>0</sub> -I/O <sub>7</sub>	Mode	Power
H	X	X	High Z	Power-Down	Standby ( $I_{\text{SB}}$ )
L	L	H	Data Out	Read	Active ( $I_{\text{CC}}$ )
L	X	L	Data In	Write	Active ( $I_{\text{CC}}$ )
L	H	H	High Z	Selected, Outputs Disabled	Active ( $I_{\text{CC}}$ )

**Read Cycle No. 1<sup>[11, 12]</sup>**



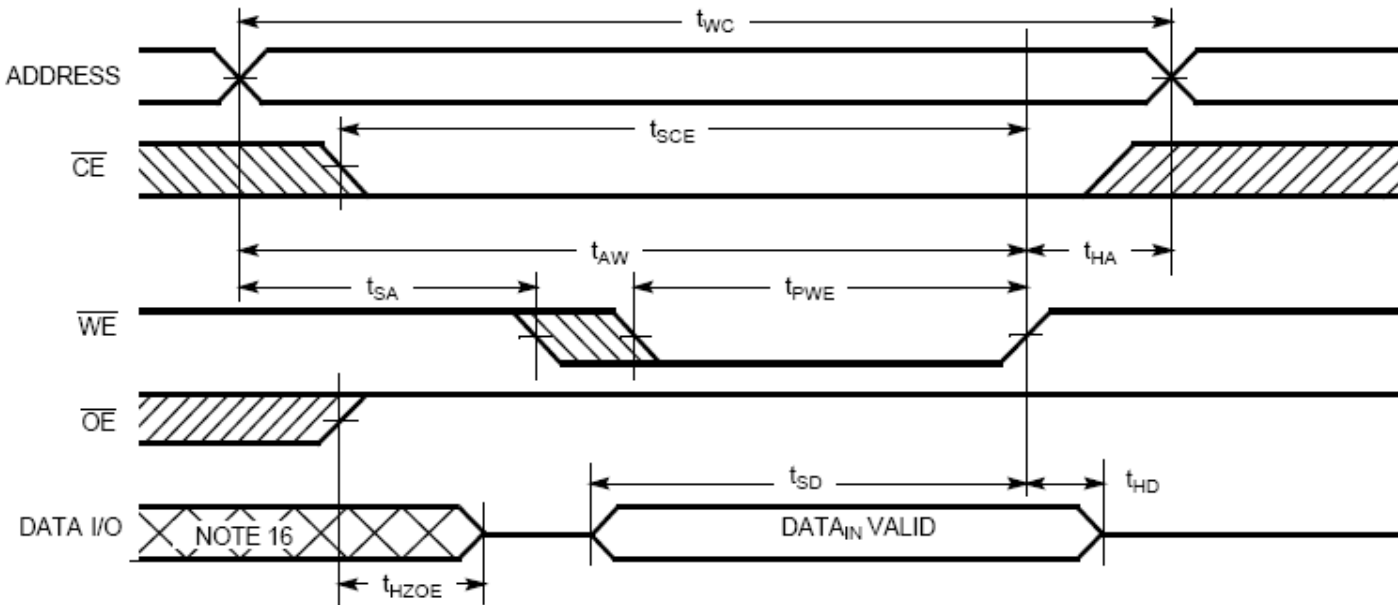
ПРИМЕР

Read Cycle No. 2 ( $\overline{OE}$  Controlled)<sup>[12, 13]</sup>Write Cycle No. 1 ( $\overline{CE}$  Controlled)<sup>[14, 15]</sup>

## Notes:

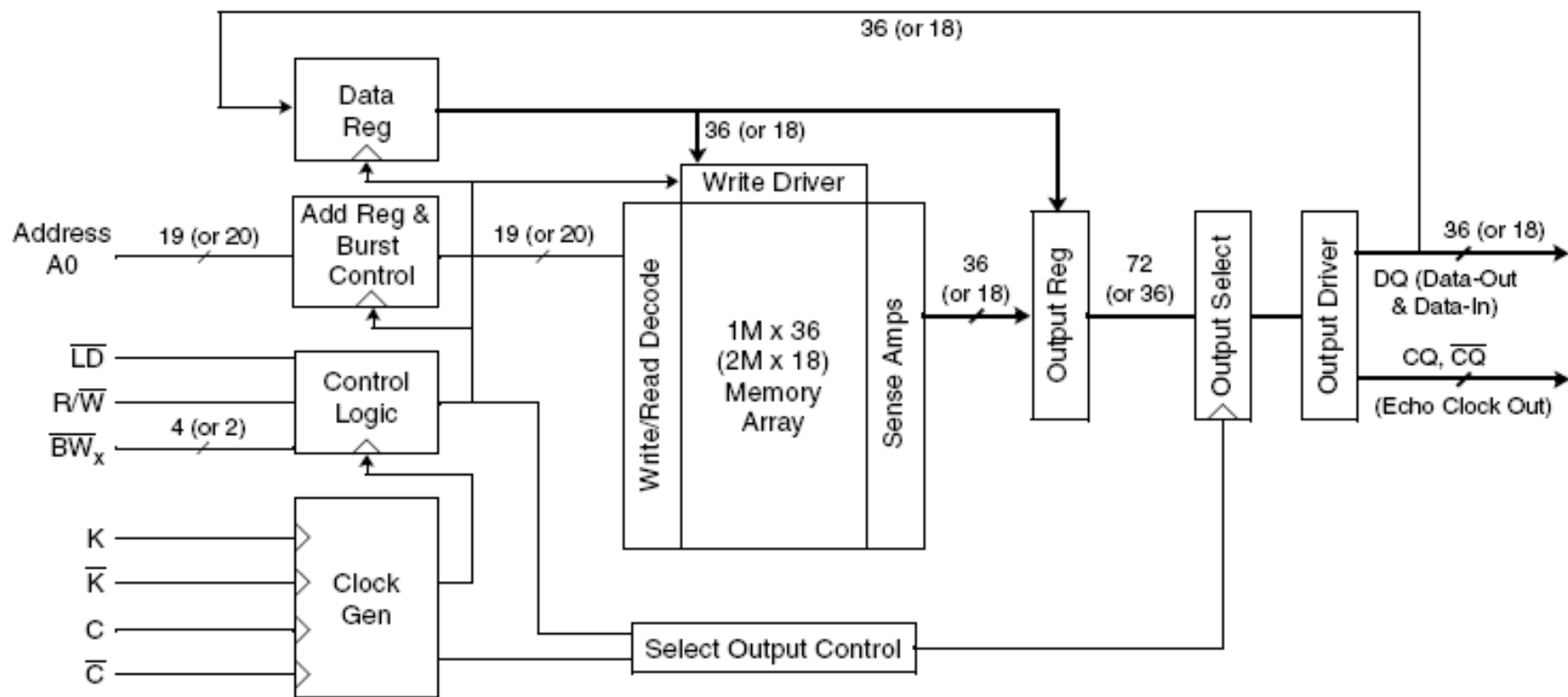
11. Device is continuously selected.  $\overline{OE}$ ,  $\overline{CE}$  =  $V_{IL}$ .
12.  $\overline{WE}$  is HIGH for read cycle.
13. Address valid prior to or coincident with  $\overline{CE}$  transition LOW.
14. Data I/O is high impedance if  $\overline{OE}$  =  $V_{IL}$ .
15. If  $\overline{CE}$  goes HIGH simultaneously with  $\overline{WE}$  going HIGH, the output remains in a high-impedance state.

Write Cycle No. 2 ( $\overline{WE}$  Controlled,  $\overline{OE}$  HIGH During Write)<sup>[14, 15]</sup>



ПРИМЕР

# 36 Mb (1M x 36 & 2M x 18) DDR-II (Burst of 2) CIO Synchronous SRAMs



ПРИМЕР

# 36 Mb (1M x 36 & 2M x 18) DDR-II (Burst of 2) CIO Synchronous SRAMs

*ISSI*<sup>®</sup>

## Features

- 1M x 36 or 2M x 18.
- On-chip delay-locked loop (DLL) for wide data valid window.
- Common data input/output bus.
- Synchronous pipeline read with self-timed late write operation.
- Double data rate (DDR-II) interface for read and write input ports.
- Fixed 2-bit burst for read and write operations.
- Clock stop support.
- Two input clocks (K and  $\bar{K}$ ) for address and control registering at rising edges only.
- Two input clocks (C and  $\bar{C}$ ) for data output control.
- Two echo clocks (CQ and  $\bar{CQ}$ ) that are delivered simultaneously with data.
- +1.8V core power supply and 1.5, 1.8V  $V_{DDQ}$ , used with 0.75, 0.9V  $V_{REF}$
- HSTL input and output levels.
- Registered addresses, write and read controls, byte writes, data in, and data outputs.
- Full data coherency.
- Boundary scan using limited set of JTAG 1149.1 functions.
- Byte write capability.
- Fine ball grid array (FBGA) package
  - 15mm x 17mm body size
  - 1mm pitch
  - 165-ball (11 x 15) array
- Programmable impedance output drivers via 5x user-supplied precision resistor.



# 36 Mb (1M x 36 & 2M x 18) DDR-II (Burst of 2) CIO Synchronous SRAMs

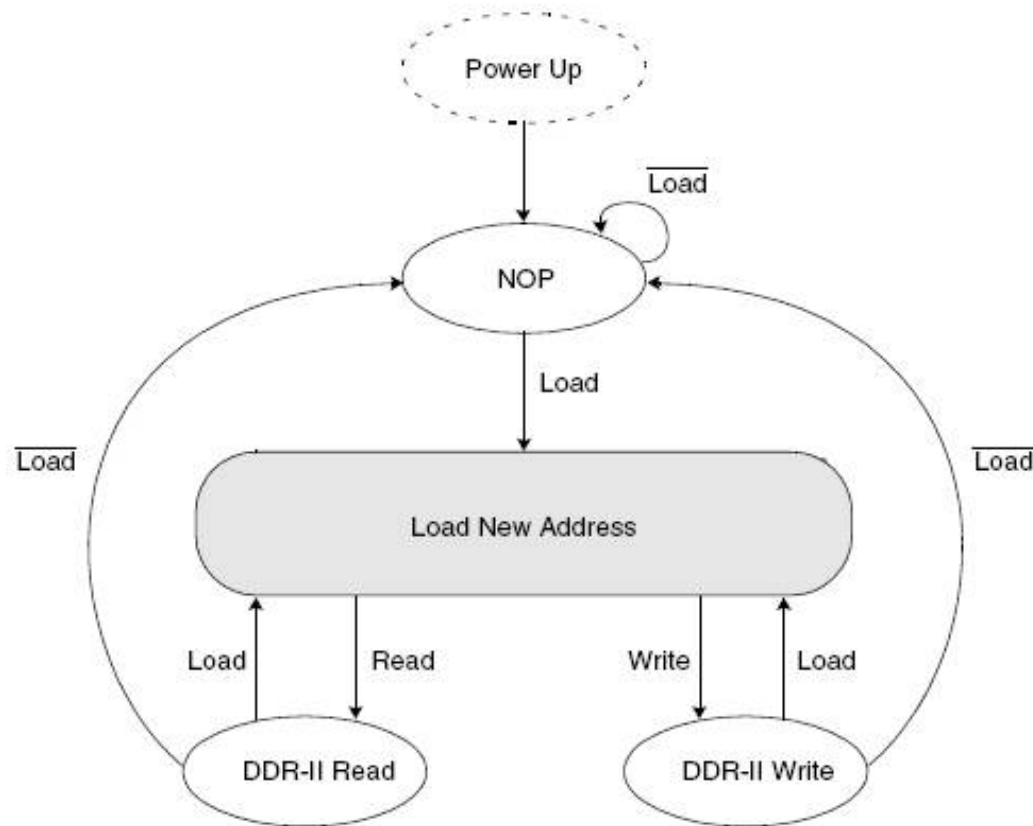


ПРИМЕР

Symbol	Pin Number	Description
$K, \bar{K}$	6B, 6A	Input clock.
$C, \bar{C}$	6P, 6R	Input clock for output data control.
$CQ, \bar{CQ}$	11A, 1A	Output echo clock.
$\bar{D}_{off}$	1H	DLL disable when low.
$SA_0$	6C	Burst count address input.
SA	9A, 4B, 8B, 5C, 7C, 5N, 6N, 7N, 4P, 5P, 7P, 8P, 3R, 4R, 5R, 7R, 8R, 9R	1M x 36 address inputs.
SA	3A, 9A, 4B, 8B, 5C, 7C, 5N, 6N, 7N, 4P, 5P, 7P, 8P, 3R, 4R, 5R, 7R, 8R, 9R	2M x 18 address inputs.
DQ0–DQ8 DQ9–DQ17 DQ18–DQ26 DQ27–DQ35	11P, 11M, 11L, 11K, 11J, 11F, 11E, 11C, 11B 10P, 11N, 10M, 10K, 10J, 11G, 10E, 11D, 10C 3B, 3D, 3E, 3F, 3G, 3K, 3L, 3N, 3P 2B, 3C, 2D, 2F, 2G, 3J, 2L, 3M, 2N	1M x 36 DQ pins
DQ0–DQ8 DQ9–DQ17	11P, 10M, 11L, 11K, 10J, 11F, 11E, 10C, 11B 2B, 3D, 3E, 2F, 3G, 3K, 2L, 3N, 3P	2M x 18 DQ pins
$R/\bar{W}$	4A	Read/write control. Read when active high.
$\bar{LD}$	8A	Synchronizes load. Loads new address when low.
$\bar{BW}_0, \bar{BW}_1, \bar{BW}_2, \bar{BW}_3$	7B, 7A, 5A, 5B	1M x 36 byte write control, active low.
$\bar{BW}_0, \bar{BW}_1$	7B, 5A	2M x 18 byte write control, active low.

# 36 Mb (1M x 36 & 2M x 18) DDR-II (Burst of 2) CIO Synchronous SRAMs

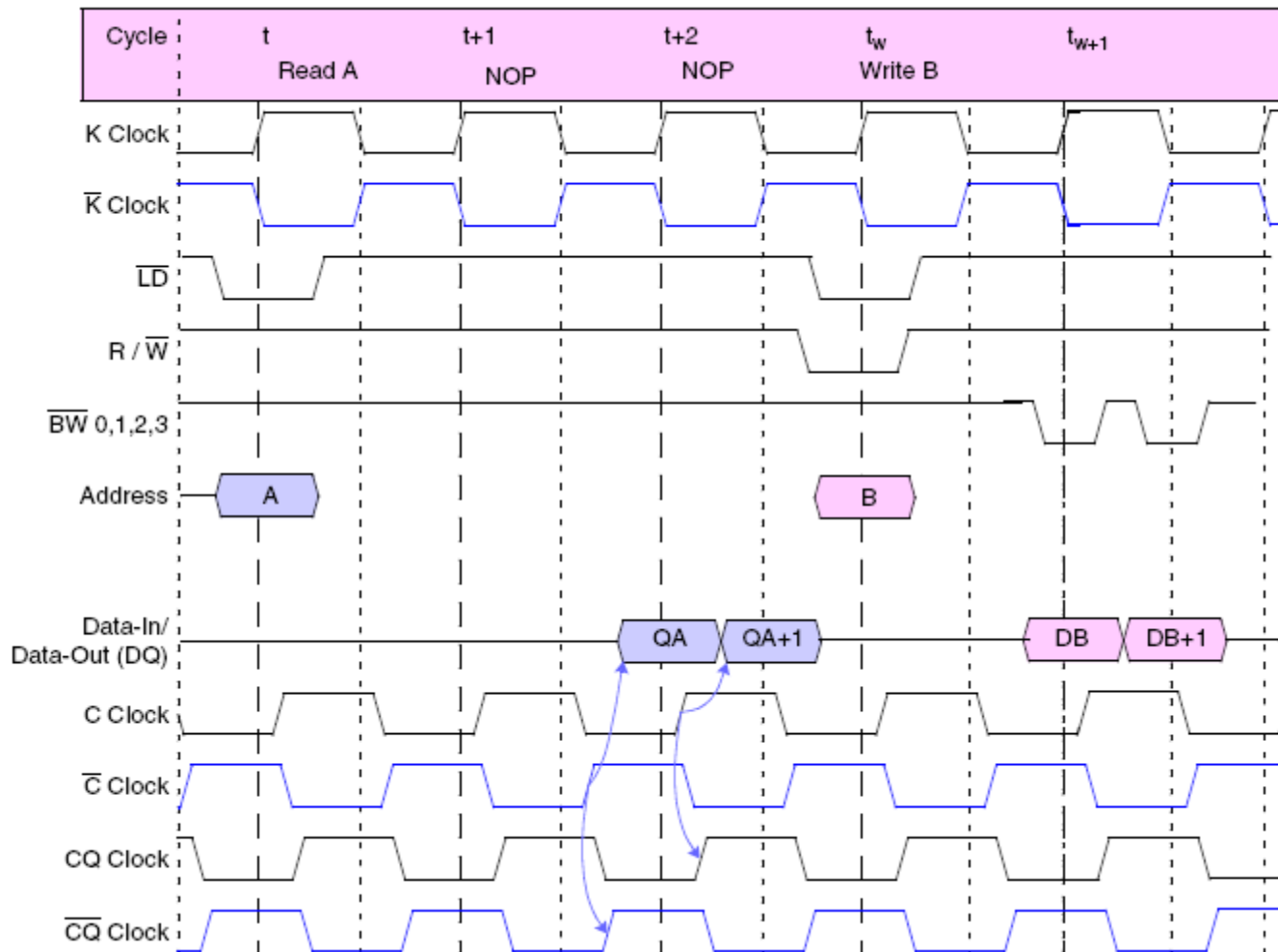
# ISSI<sup>®</sup>



- Notes:**
1. Internal burst counter is fixed as two-bit linear; that is, when first address is  $A0+0$ , next internal burst address is  $A0+1$ .
  2. *Read* refers to read active status with  $R/\overline{W}$  = high.
  3. *Write* refers to write active status with  $R/\overline{W}$  = low.
  4. *Load* refers to read new address active status with  $\overline{LD}$  = low.
  5. *Load* is read new address inactive status with  $\overline{LD}$  = high.

# 36 Mb (1M x 36 & 2M x 18) DDR-II (Burst of 2) CIO Synchronous SRAMs

**ISSI**<sup>®</sup>



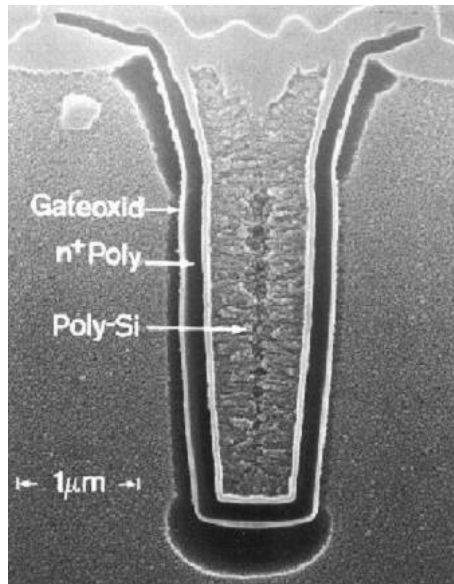
ПРИМЕР

# Динамические ЗУ с произвольной выборкой (DRAM)

DRAM для обращения по произвольным адресам  
DRAM, RDRAM

DRAM, оптимизированные для обращения по последовательным адресам:

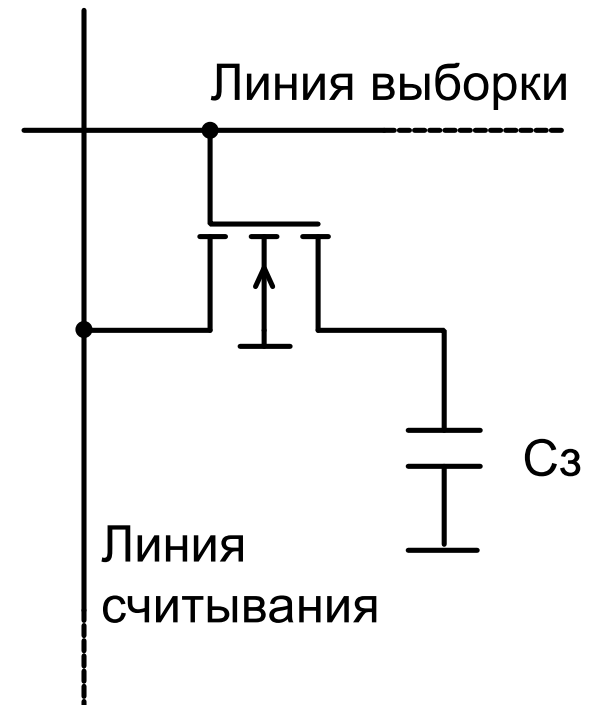
FPM DRAM, EDO DRAM, BEDO DRAM, SDRAM, DDR SDRAM, RDRAM



Архитектура ЭВМ

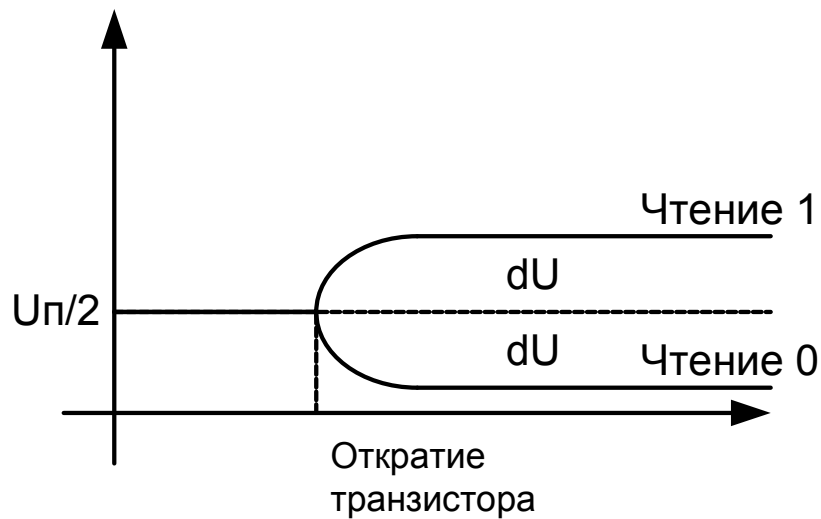
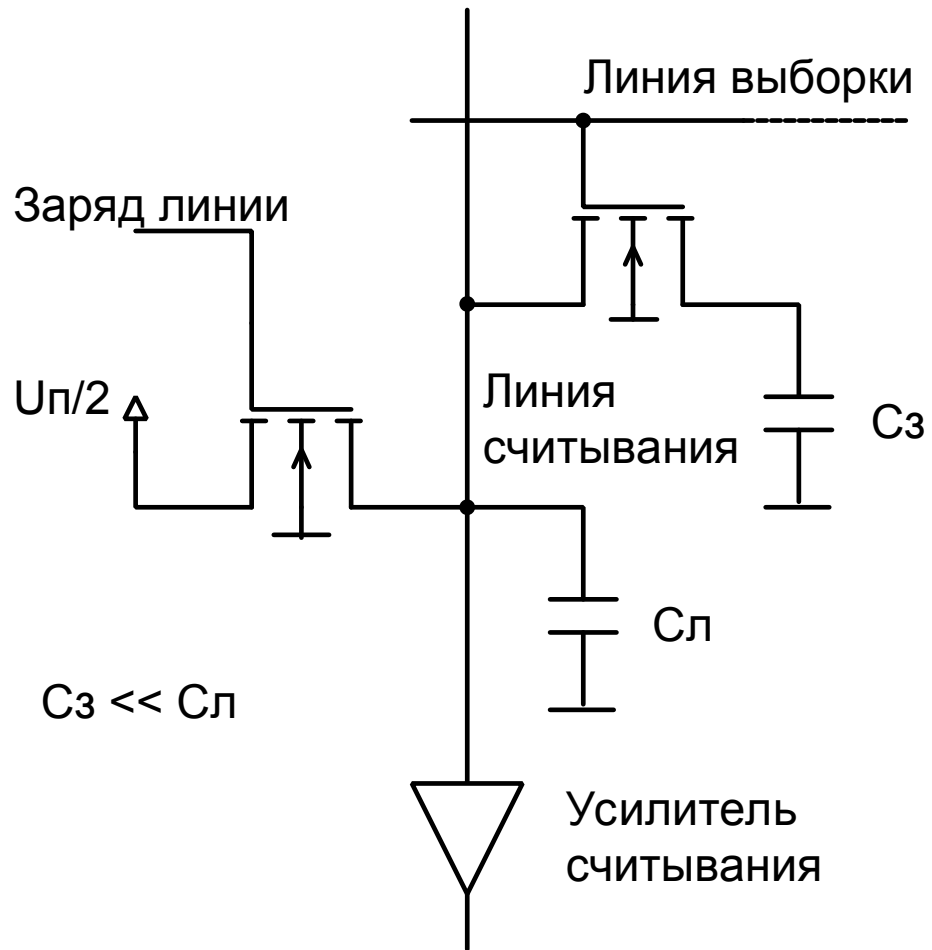
При выборке строки все  $C_3$  подключаются к линиям считывания.  
После считывания необходимо произвести обратную запись информации – регенерацию.

ИУ6

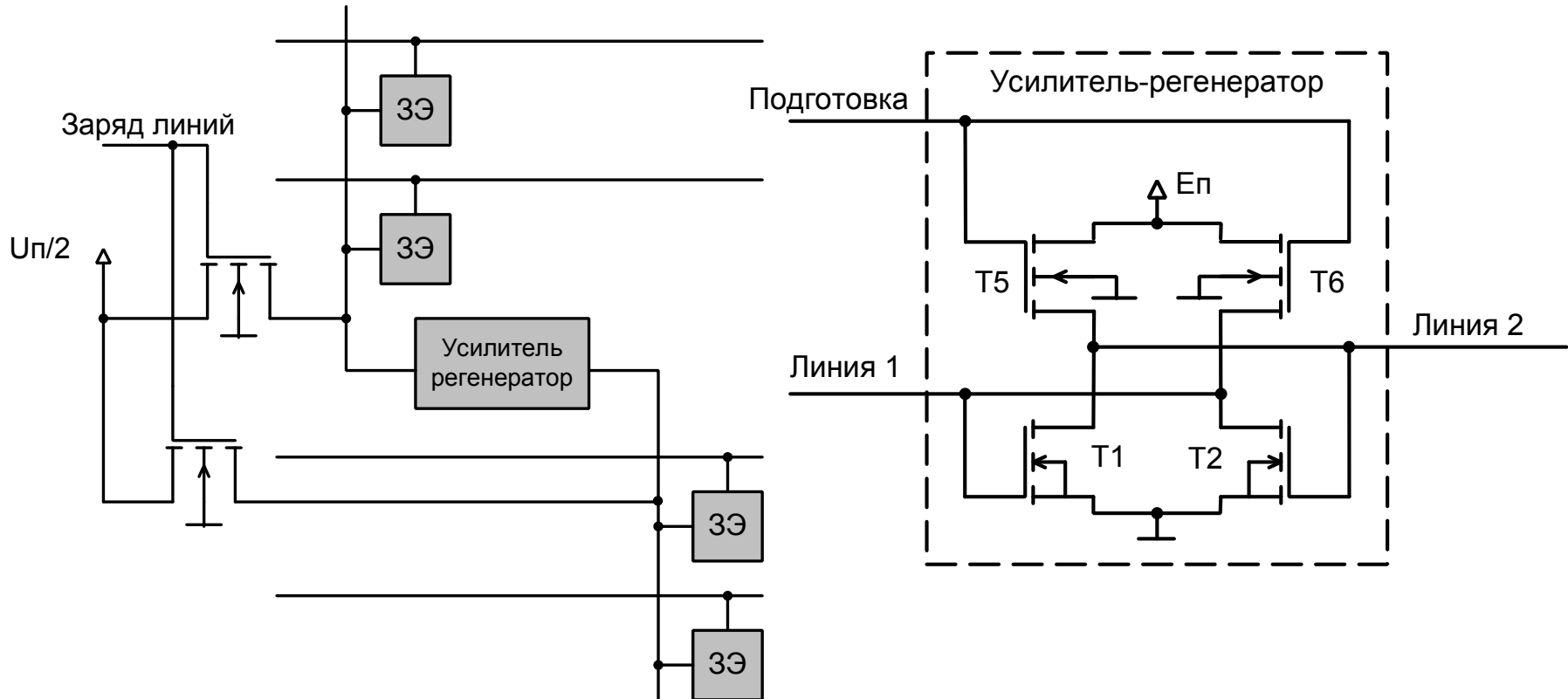


28

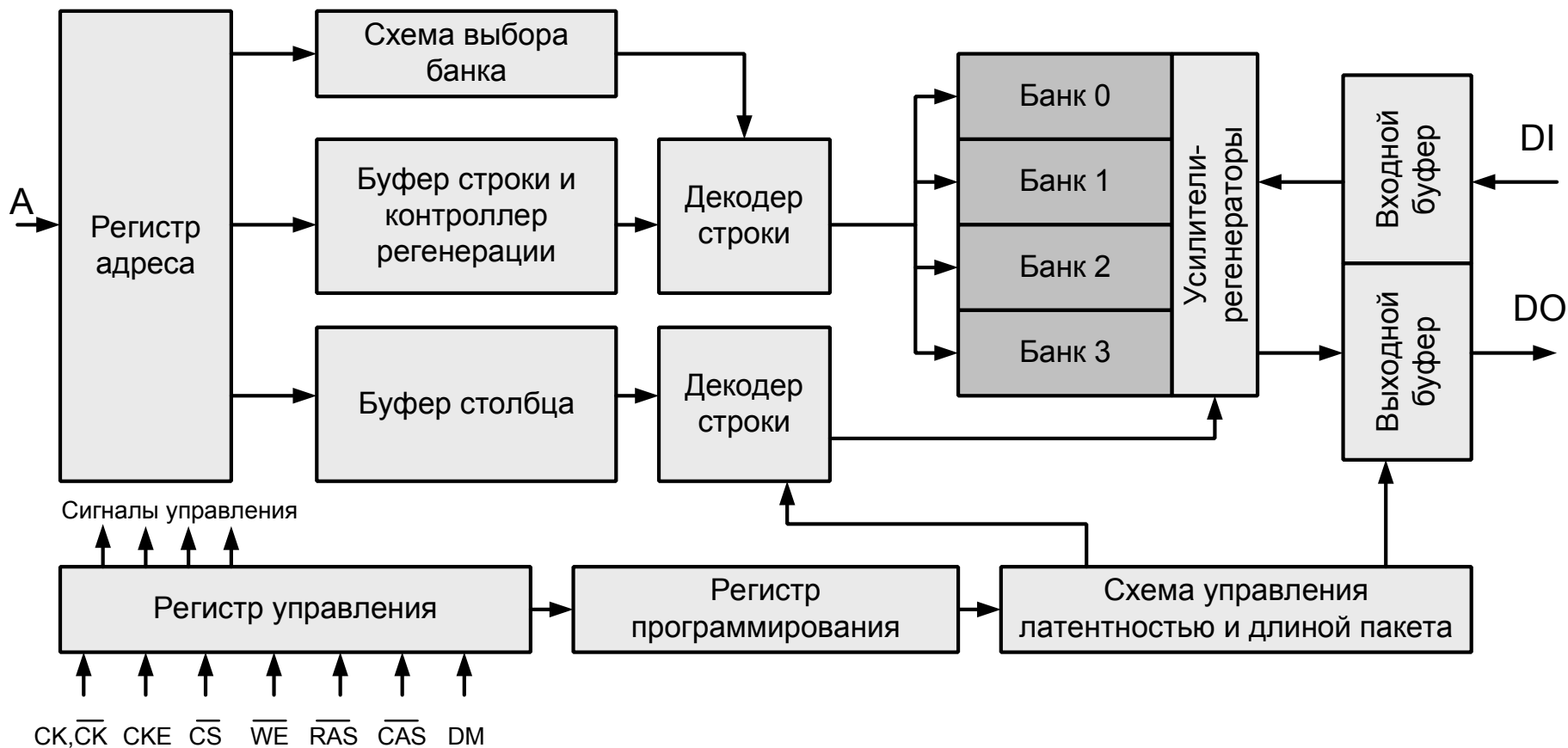
# Процесс считывания в DRAM

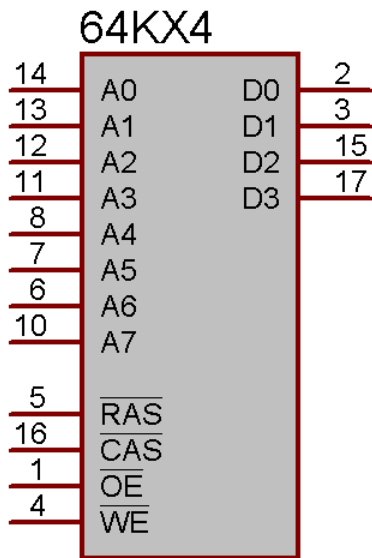


# Принцип действия усилителя-регенератора



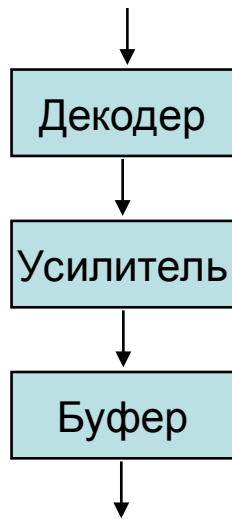
# Микросхема динамической памяти





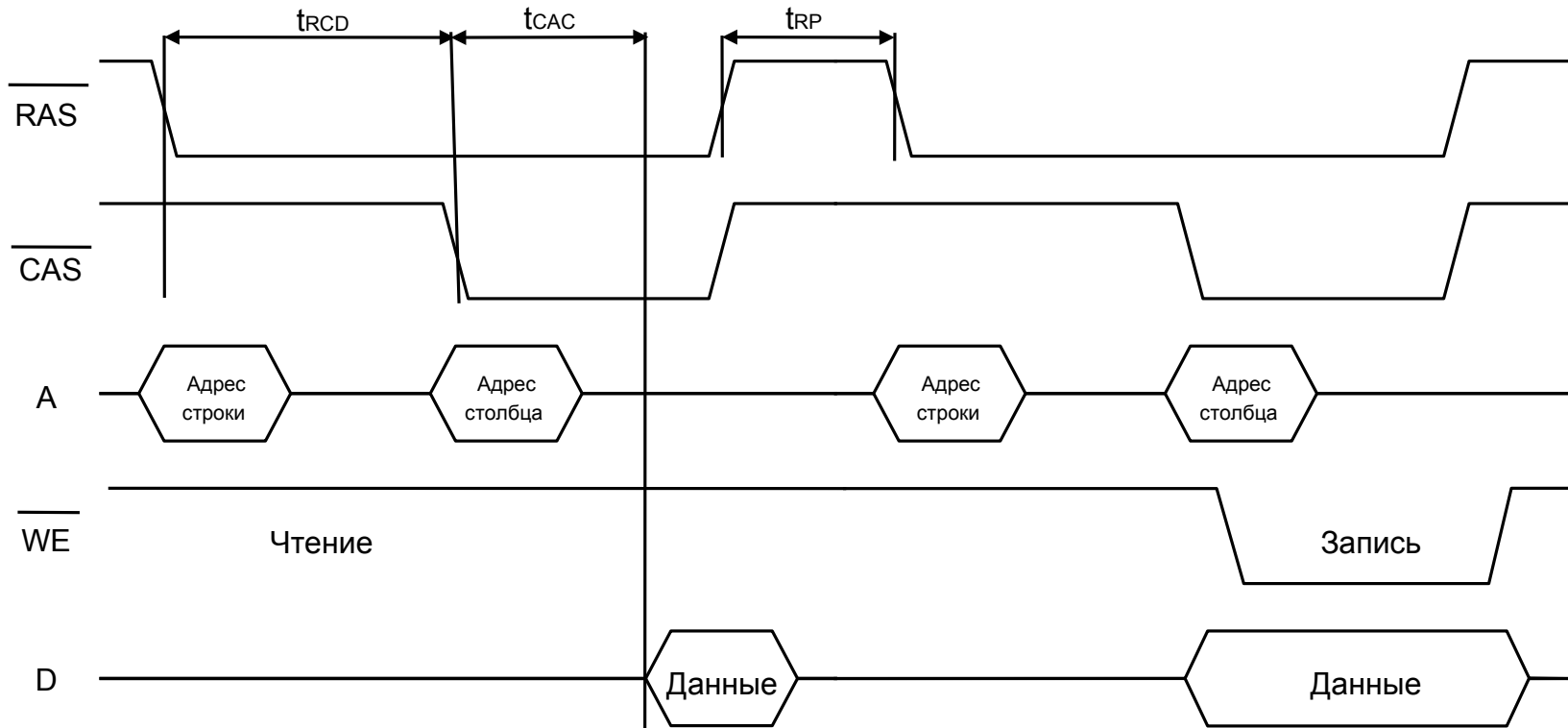
## Функциональные возможности SDRAM памяти:

- Многобанковая организация.
- Командный режим работы.
- Команды пакетного чтения/записи.
- Использование чередования банков при последовательном увеличении адресов.
- Команды пакетного чтения/записи с авто-подзарядом.
- Возможность останова чтения/записи по режиму регенерации.
- Возможность останова чтения/записи по новому запросу чтения/записи.
- Управление маскированием шины данных по сигналу DQM.
- Минимальное время (1 CLK) между последовательными командами.
- Команда PrechargeAll.
- CAS латентность 2 и 3 CLK.
- Длина пакета 1,2 и 4 слова.
- Команда само-регенерации.
- Режим энергосбережения.





# Диаграмма работы DRAM памяти

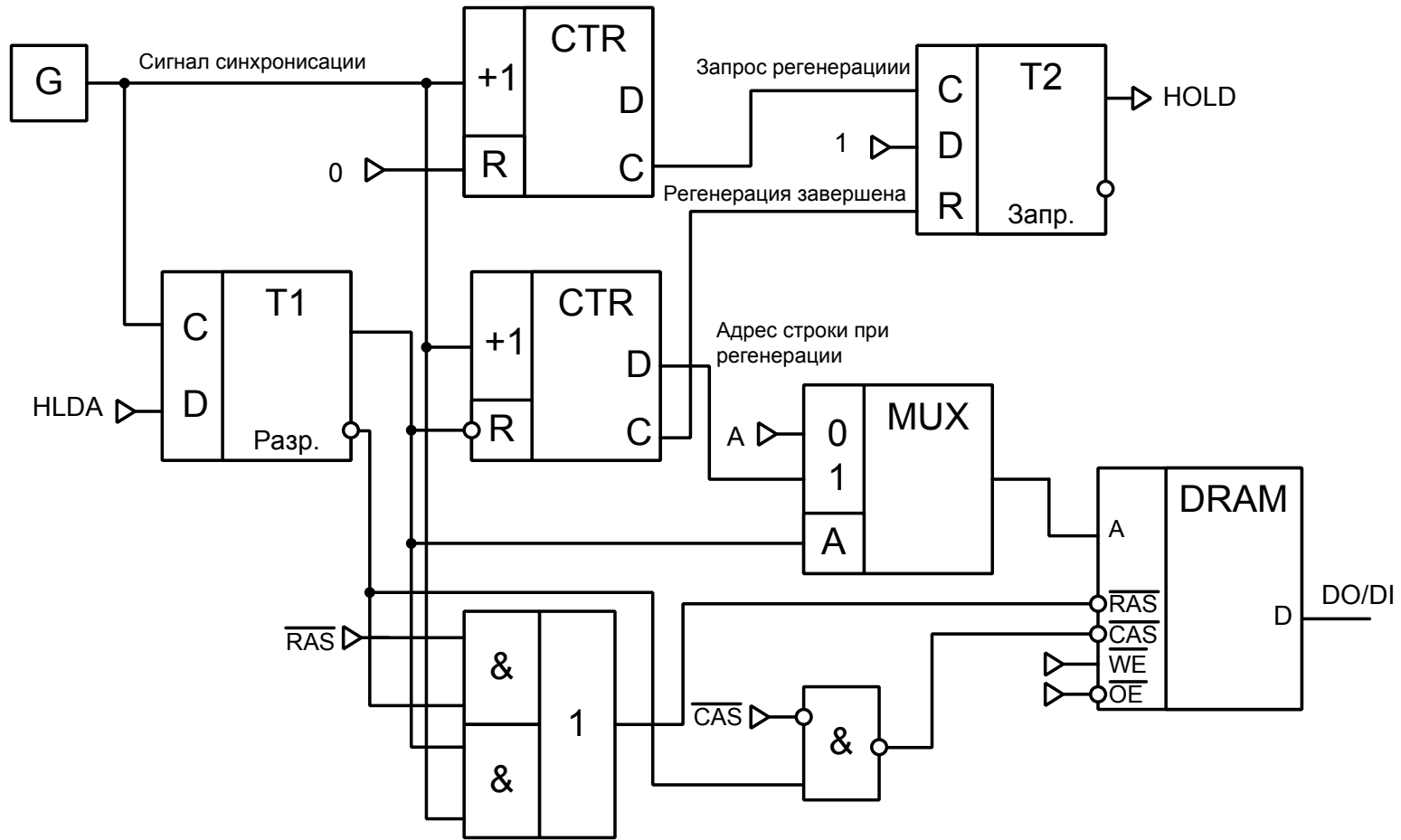


$t_{RCD}$  – RAS to CAS Delay.

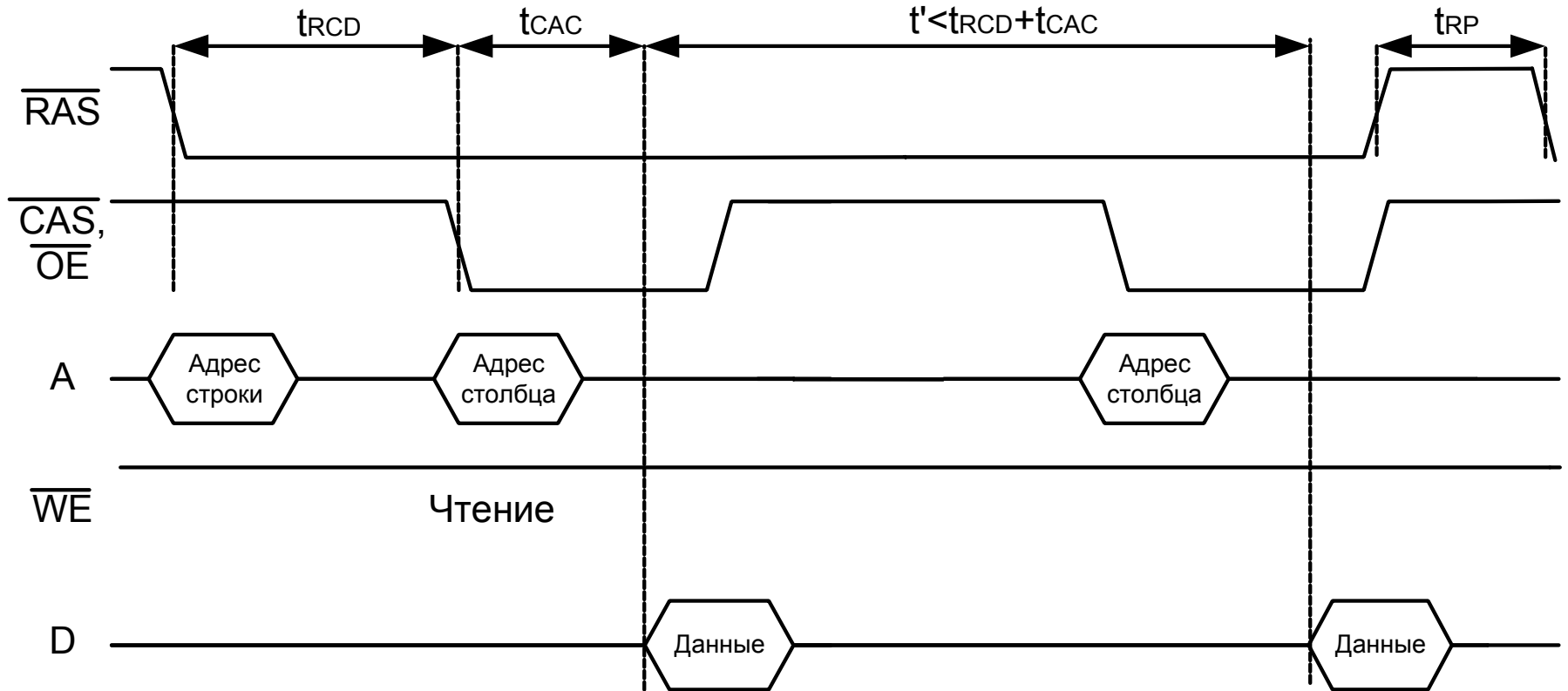
$t_{RP}$  – RAS Precharge.

$t_{CAC}$  – CAS Delay.

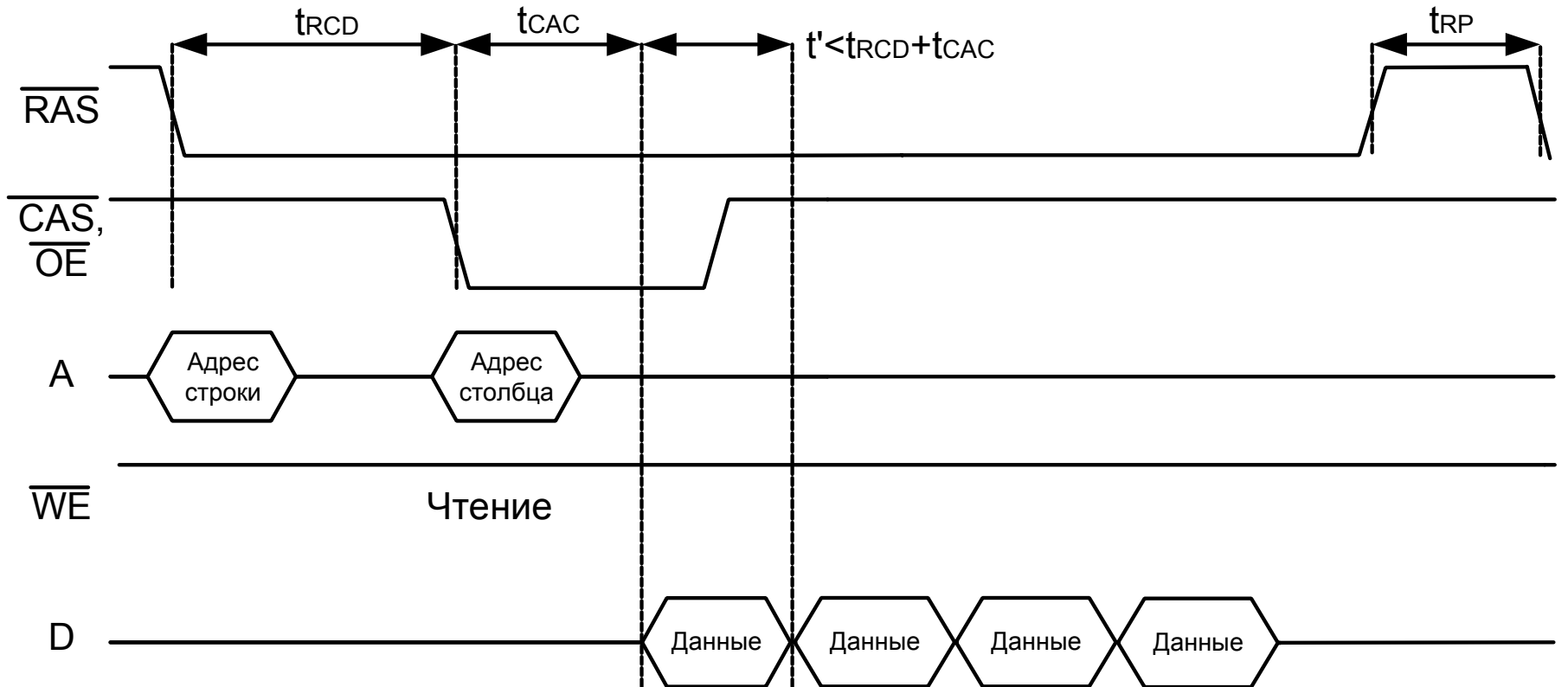
# Контроллер динамической памяти



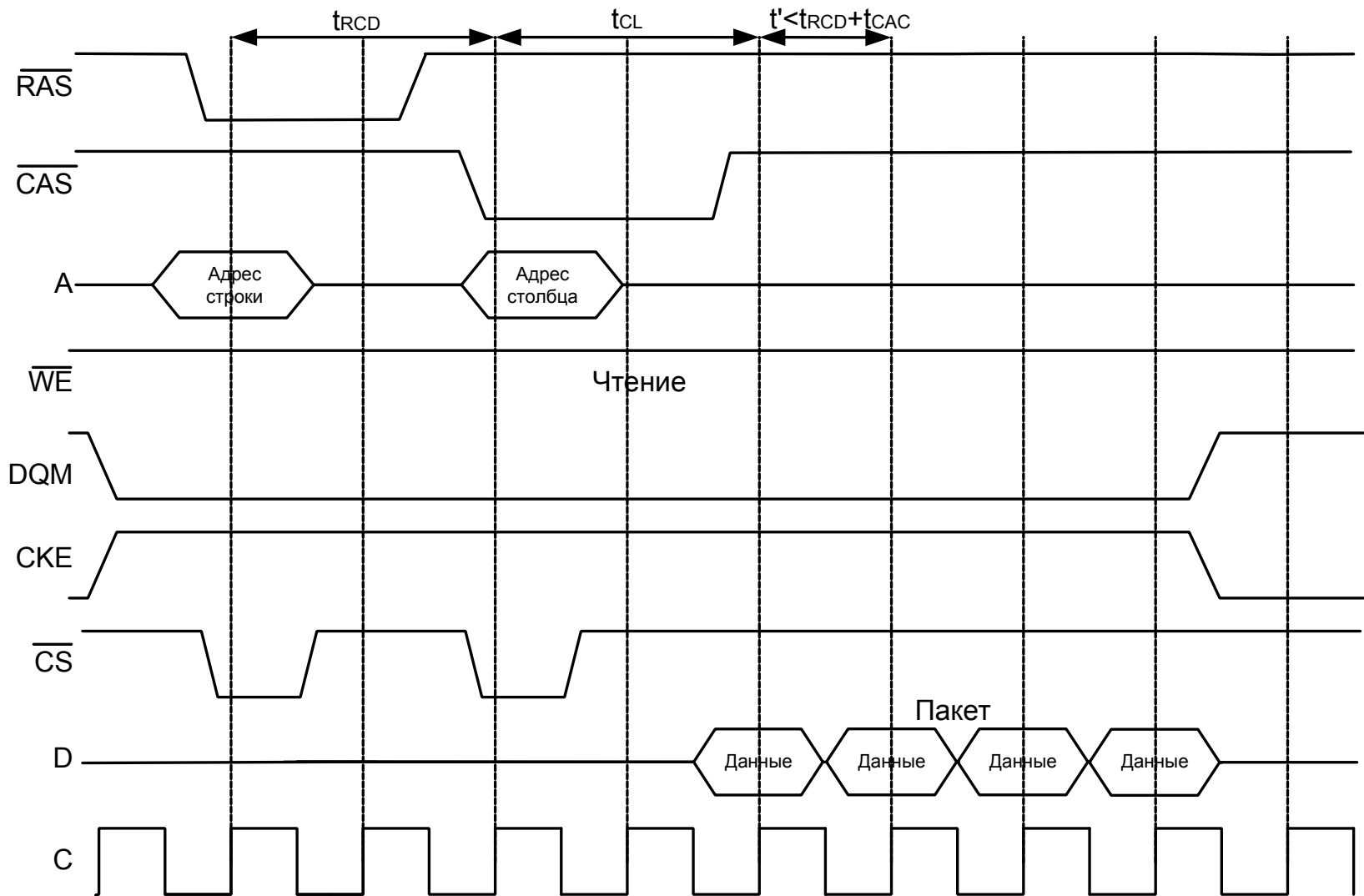
# Диаграмма работы FPM DRAM памяти



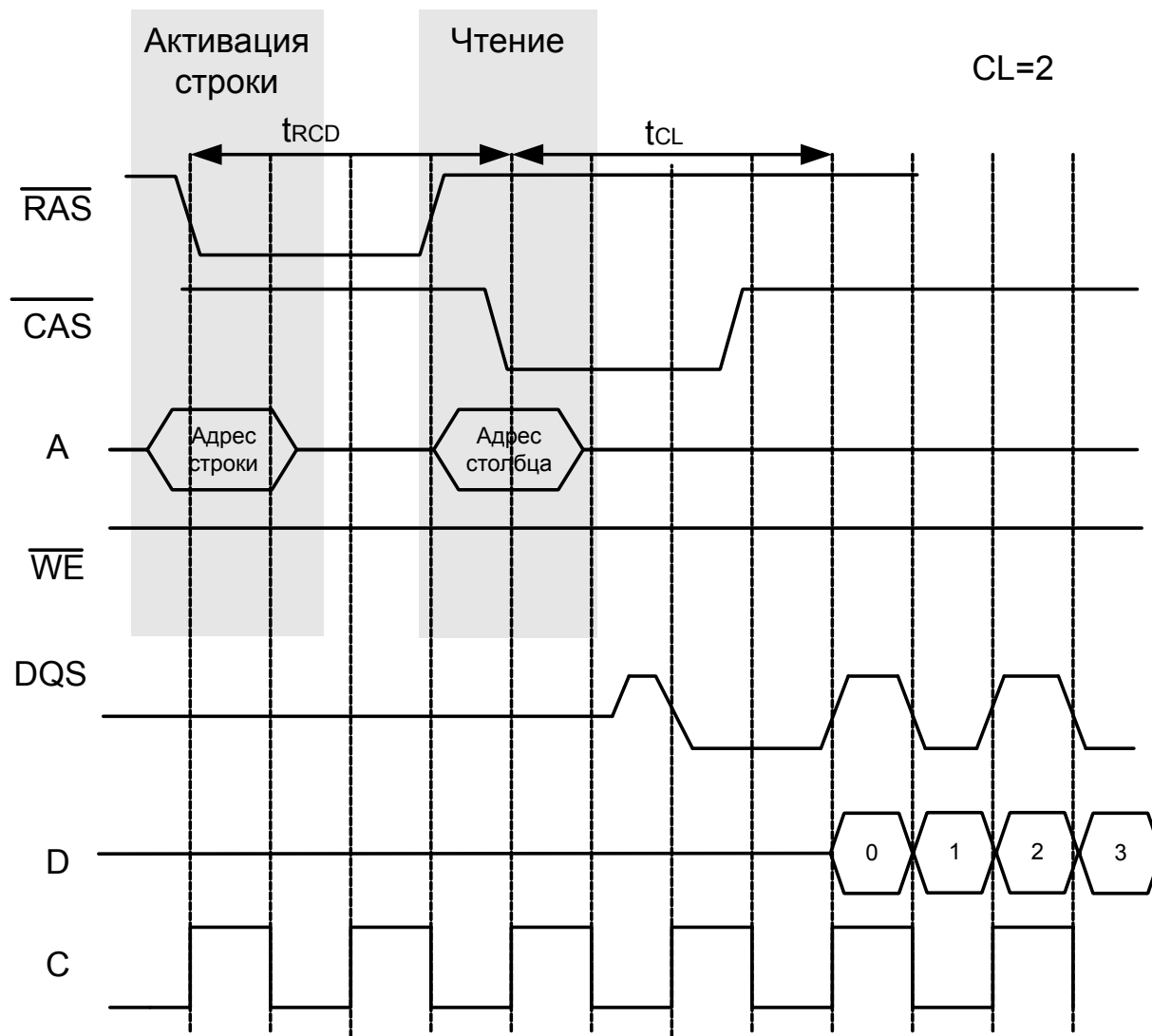
# Диаграмма работы BEDO DRAM памяти



# Диаграмма работы SDRAM памяти

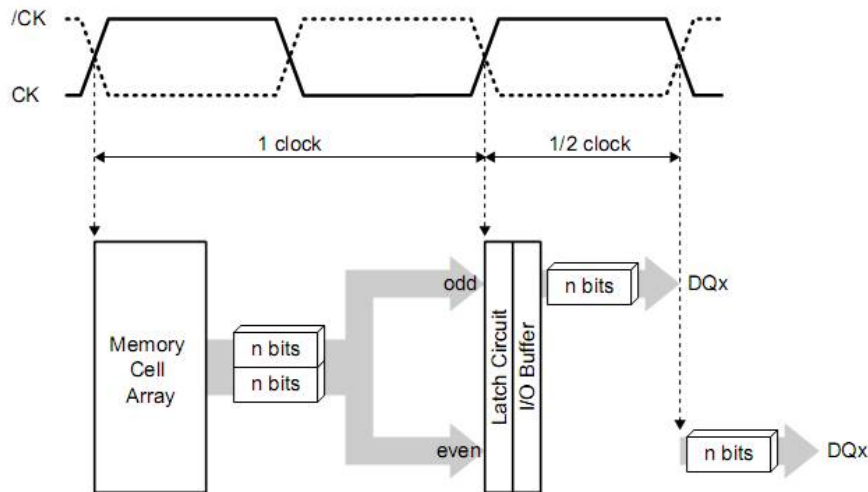


# Диаграмма работы DDR SDRAM памяти



# Способы повышения производительности RAM

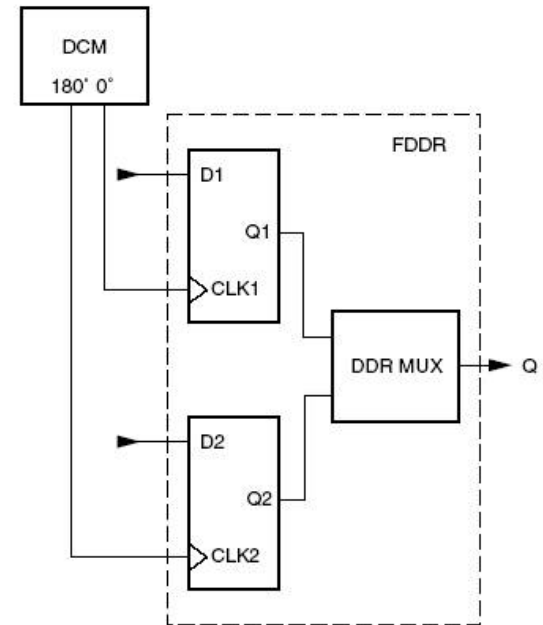
- Синхронизация.
- Конвейеризация.
- Пакетный режим обмена.
- Ускорение реверса шины.
- Чередование банков при обращении по последовательным адресам.
- Удвоение скорости.



Архитектура ЭВМ

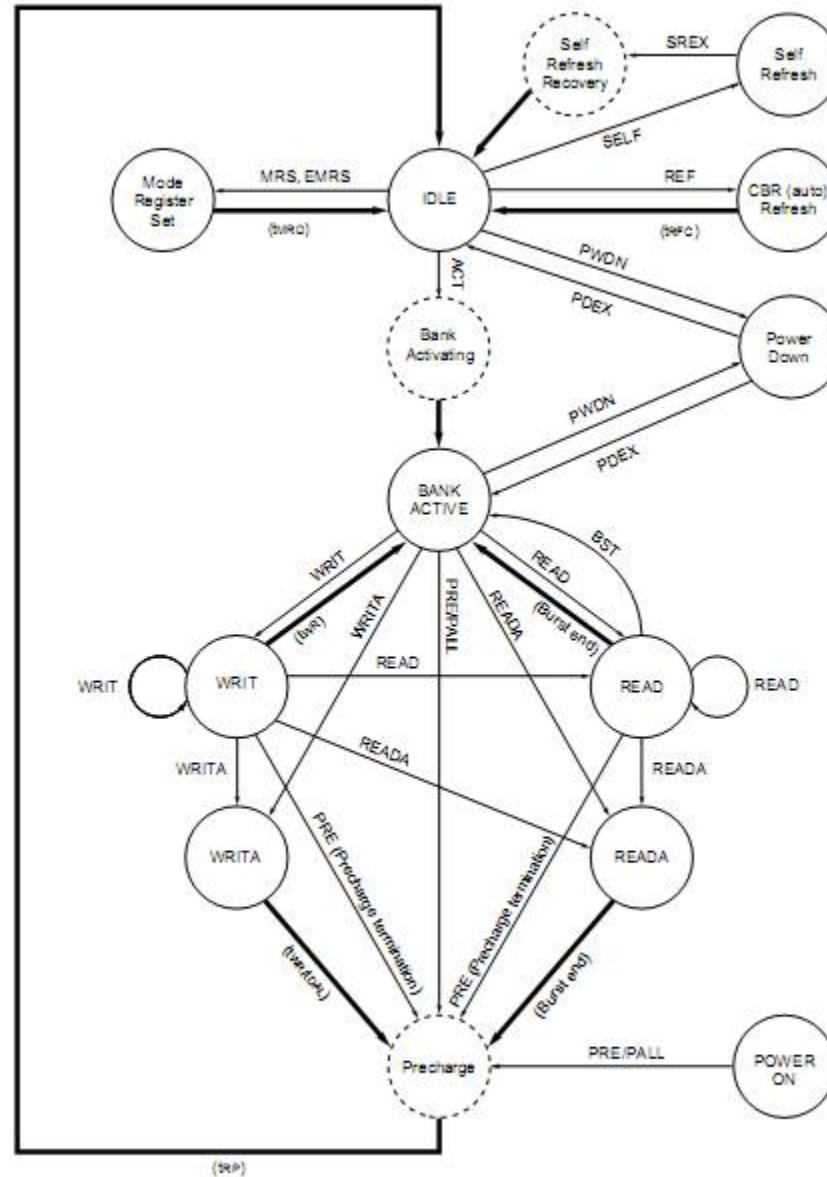
ИУ6

## Регистр DDR



39

# Диаграмма состояний YA DDR SDRAM



ПРИМЕР

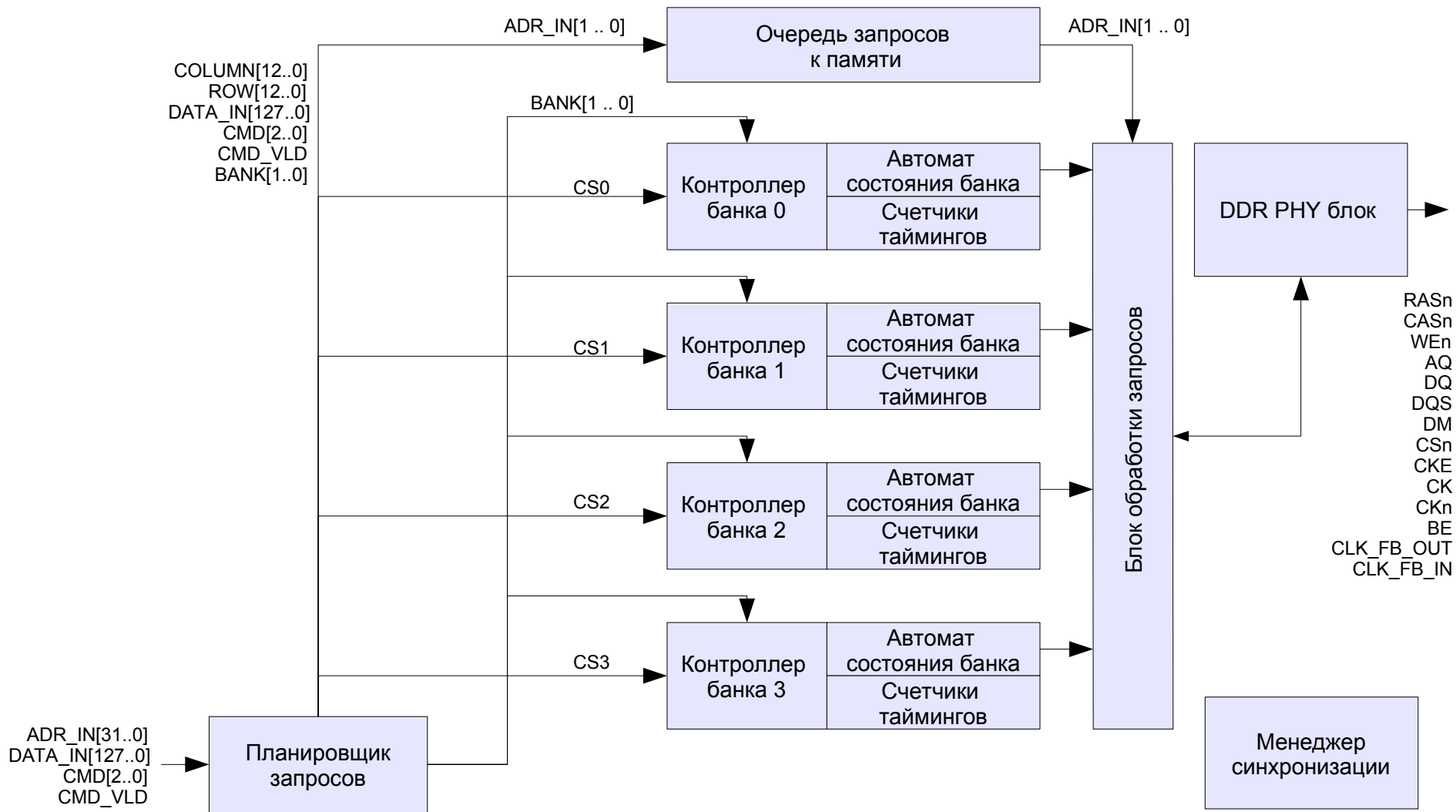
Архитектура ЭЭ

Document No. E0234E30 (Ver.3.0)  
 Date Published April 2002 (K) Japan  
 URL: <http://www.elpida.com>

 Automatic sequence  
 Manual input



# Контроллер DDR/DDR2

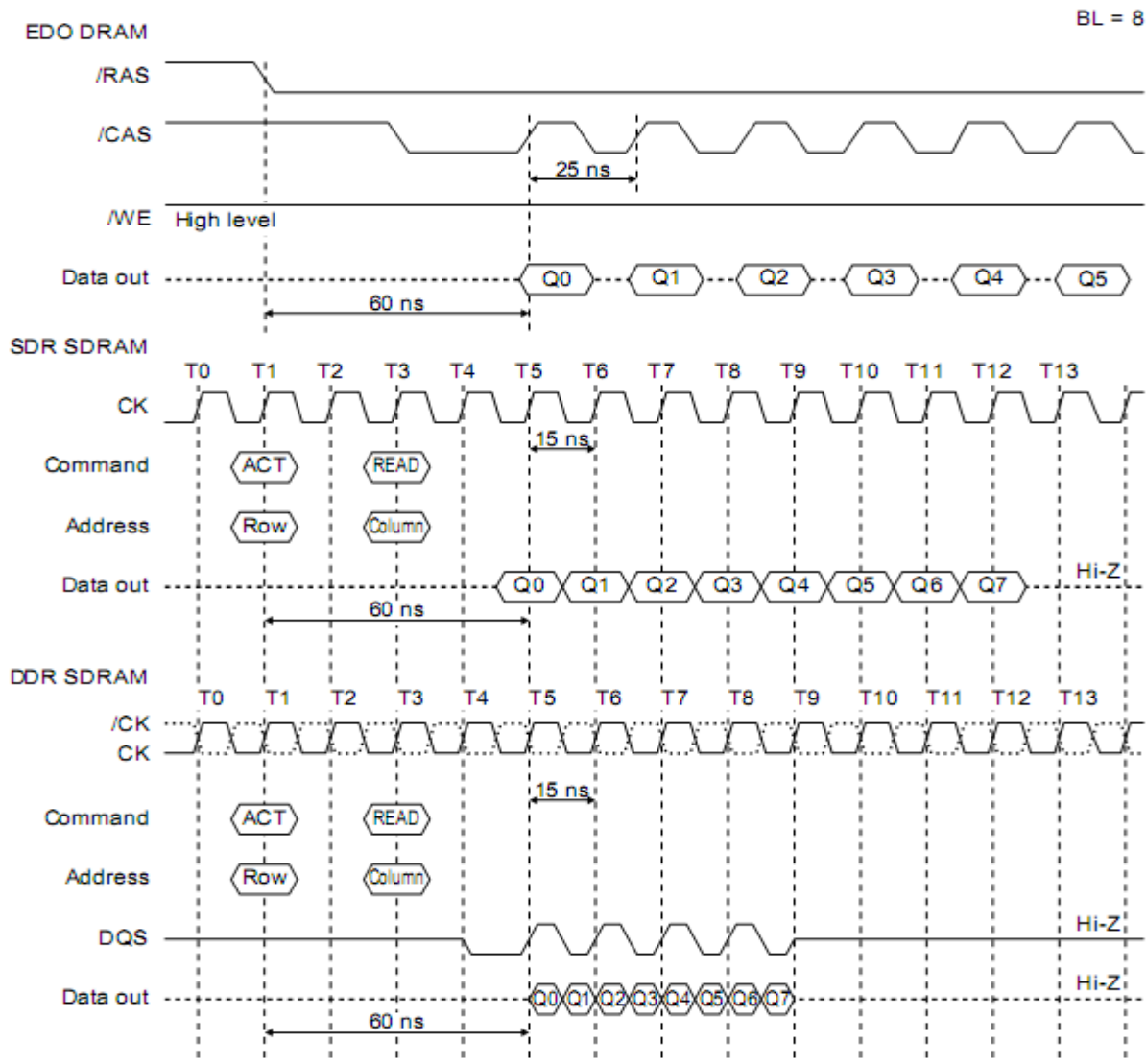


Архитектура ЭВМ

ИУ6

41

# Сравнение EDO RAM, SDRAM, DDR SDRAM



ПРИМЕР

# Сравнение DDR и DDR2

DDR память

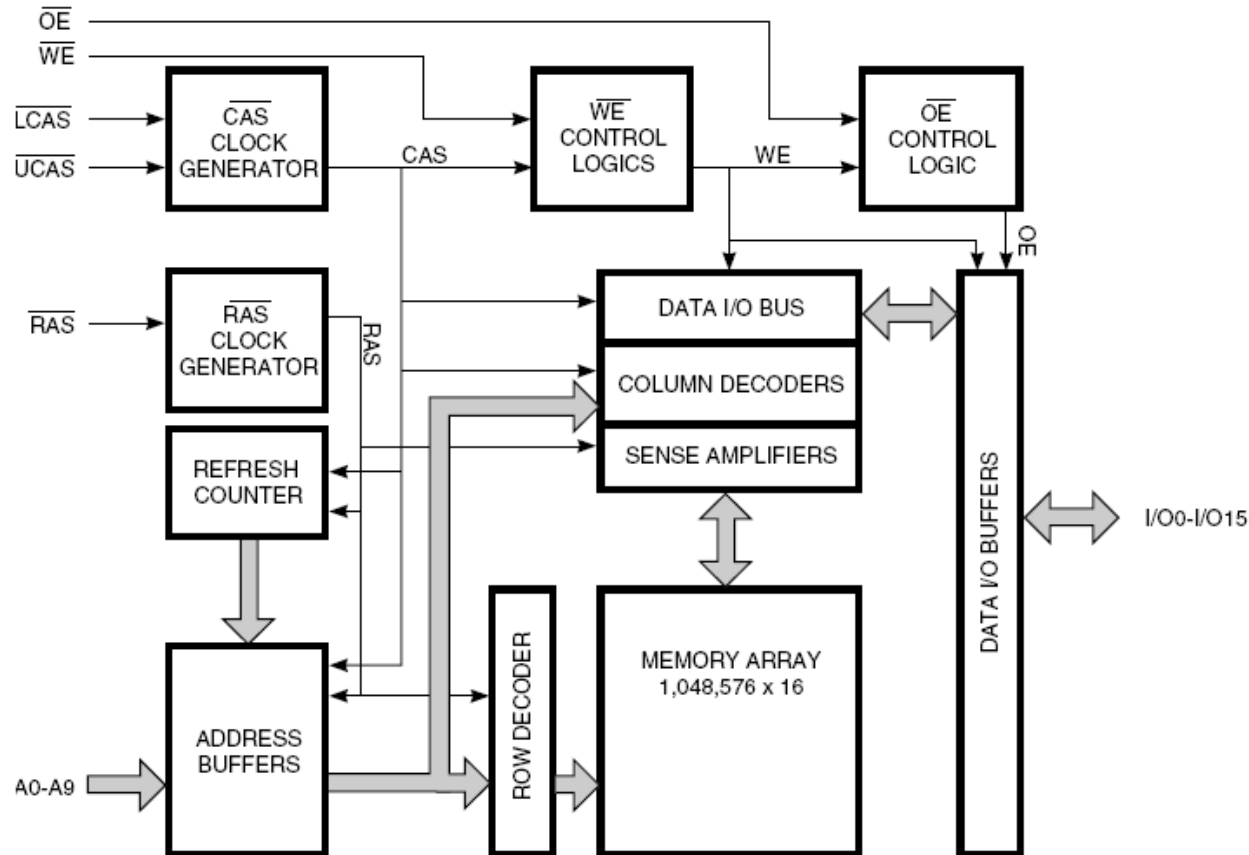
DDR2 память

# IS41C16100 IS41LV16100



## 1M x 16 (16-MBIT) DYNAMIC RAM WITH EDO PAGE MODE

December 2005



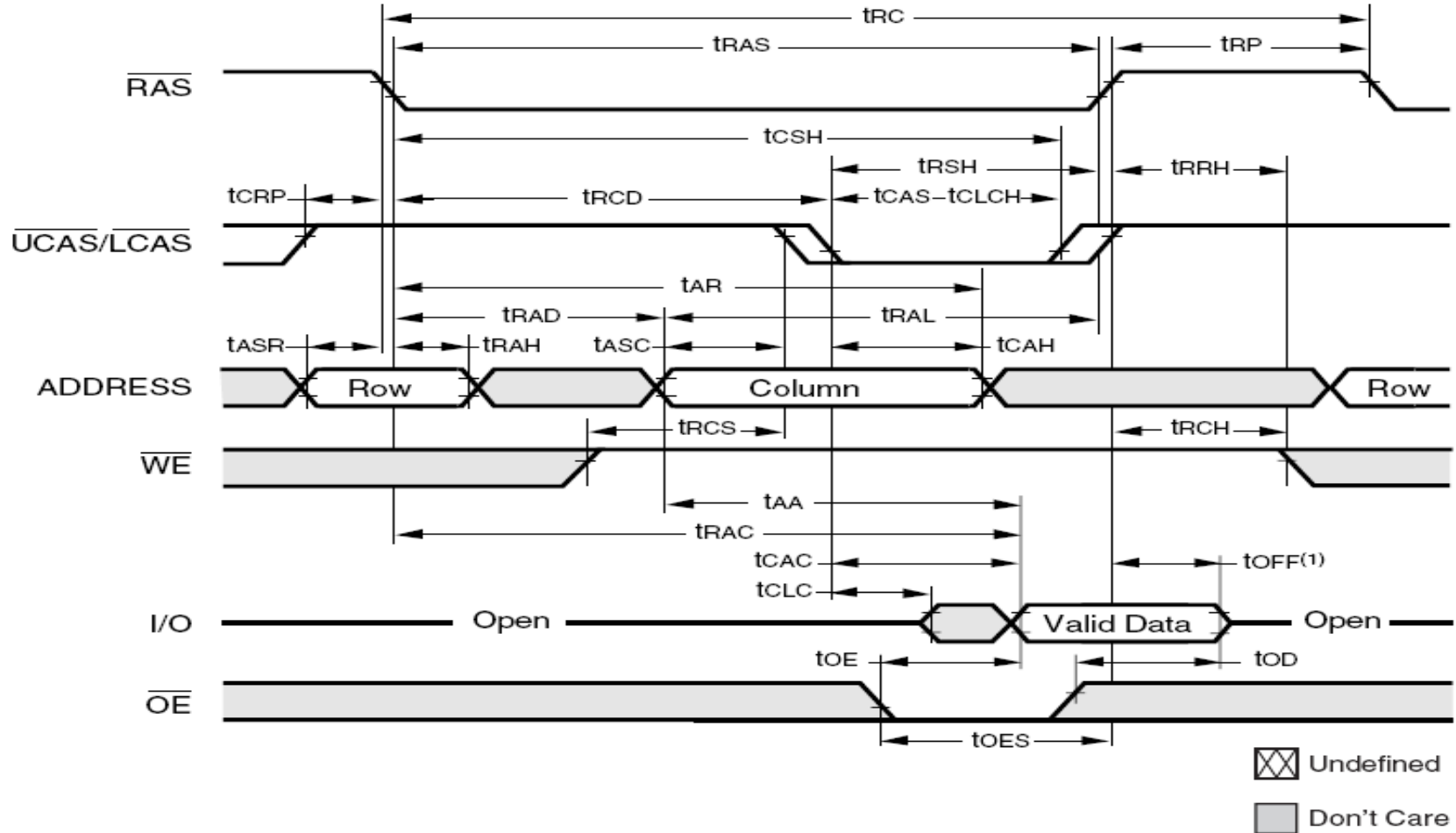
ПРИМЕР

# IS41C16100 IS41LV16100



## 1M x 16 (16-MBIT) DYNAMIC RAM WITH EDO PAGE MODE

December 2005



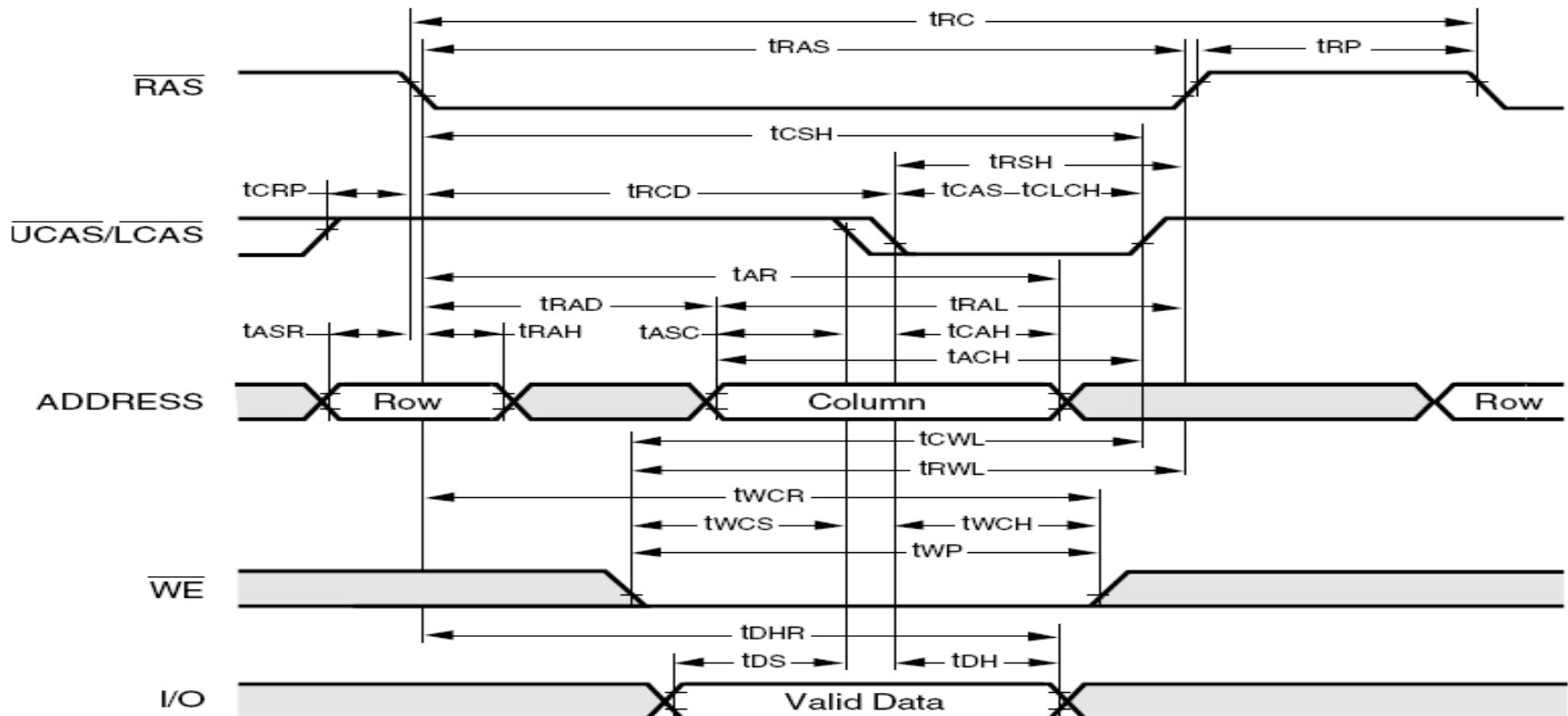
ПРИМЕР

# IS41C16100 IS41LV16100



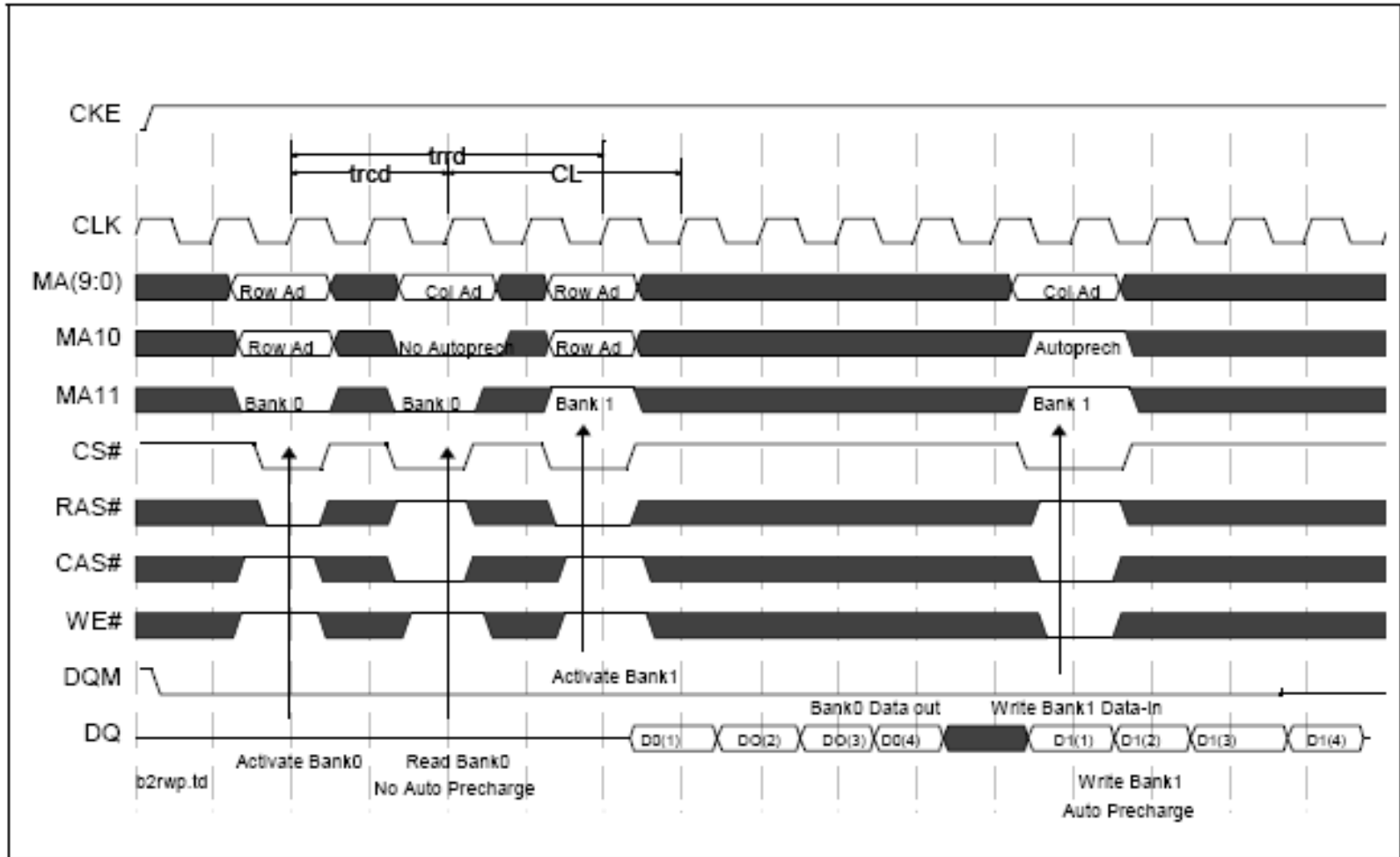
## 1M x 16 (16-MBIT) DYNAMIC RAM WITH EDO PAGE MODE

December 2005



Don't Care

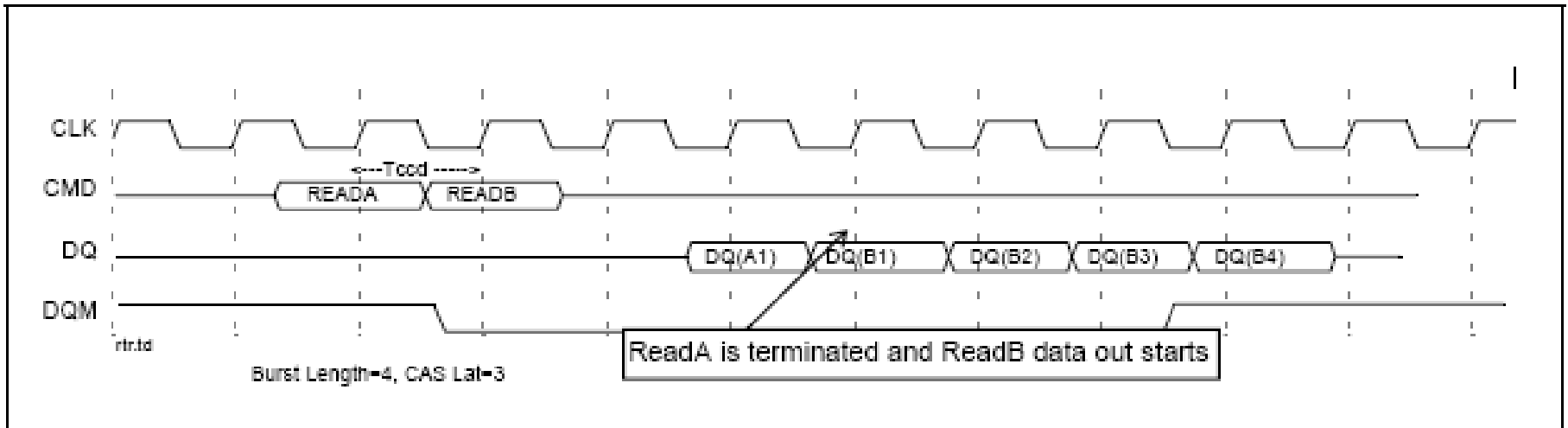
Read and Write Commands (Burst Length 4 Shown)



ПРИМЕР

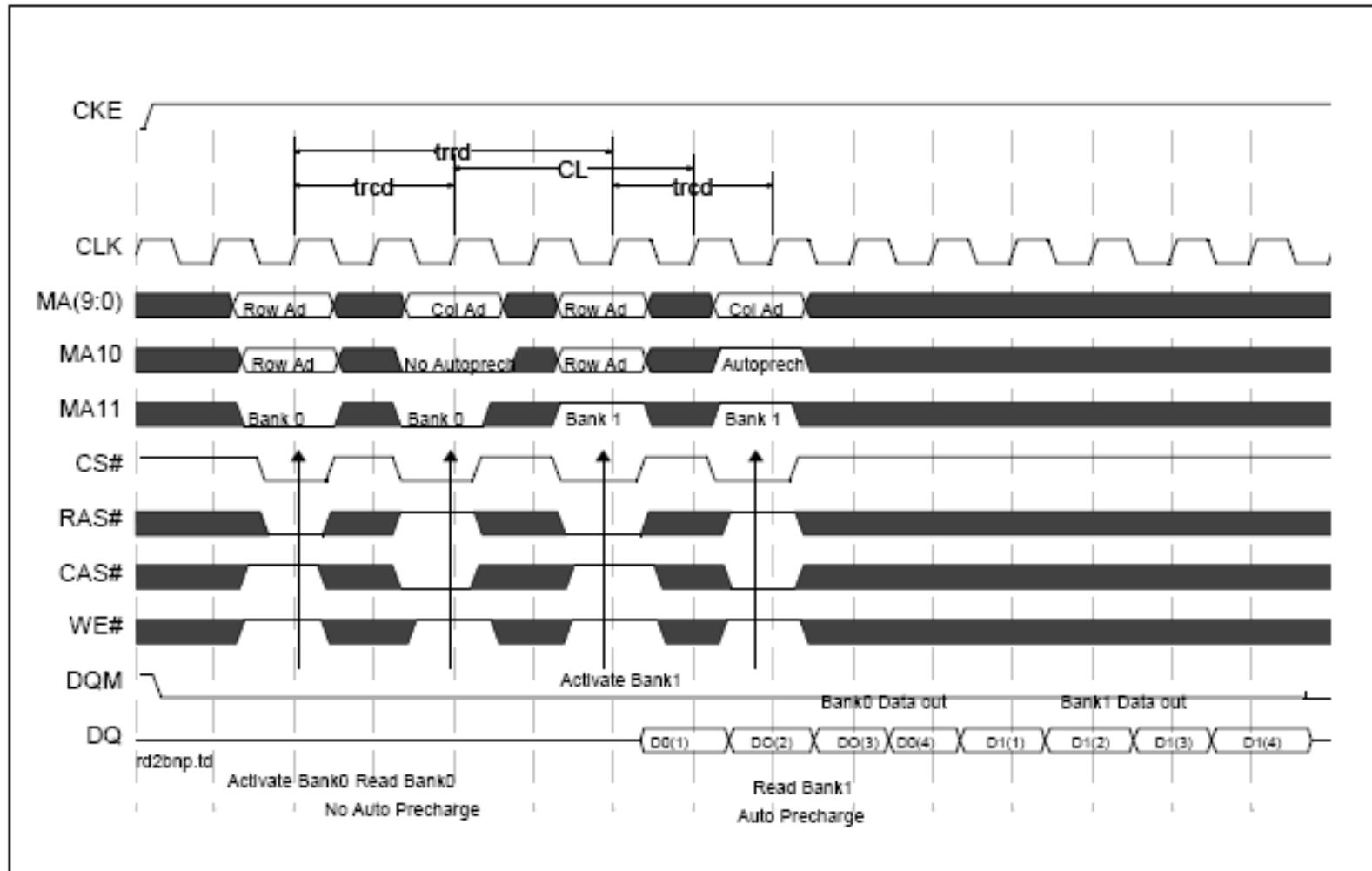
Read Terminated By Read

ПРИМЕР





### Two Bank Ping Pong Read



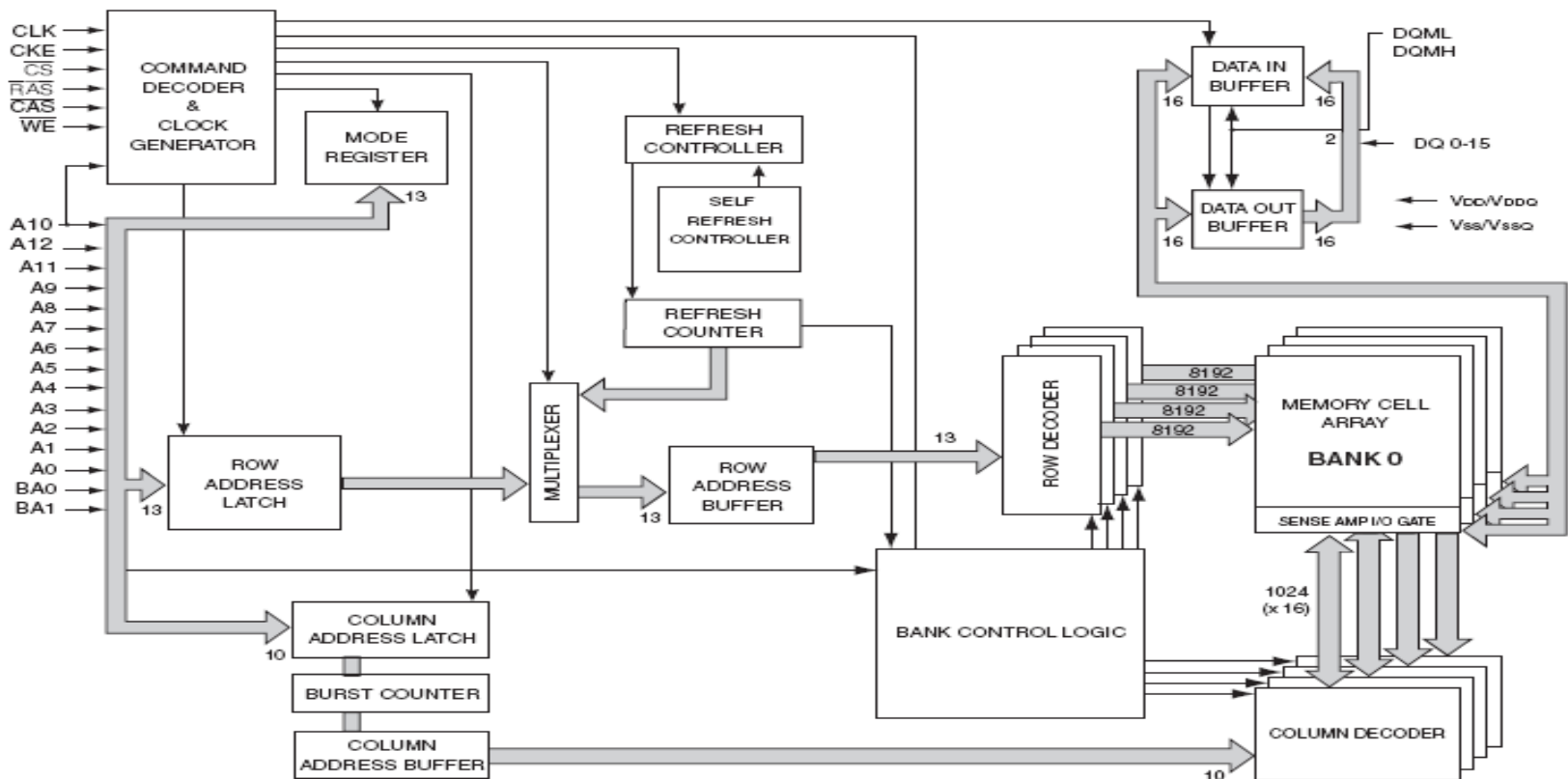
ПРИМЕР



# IS42S16320B

## 32Meg x 16 512-MBIT SYNCHRONOUS DRAM

PRELIMINARY INFORMATION  
JULY 2007



ПРИМЕР

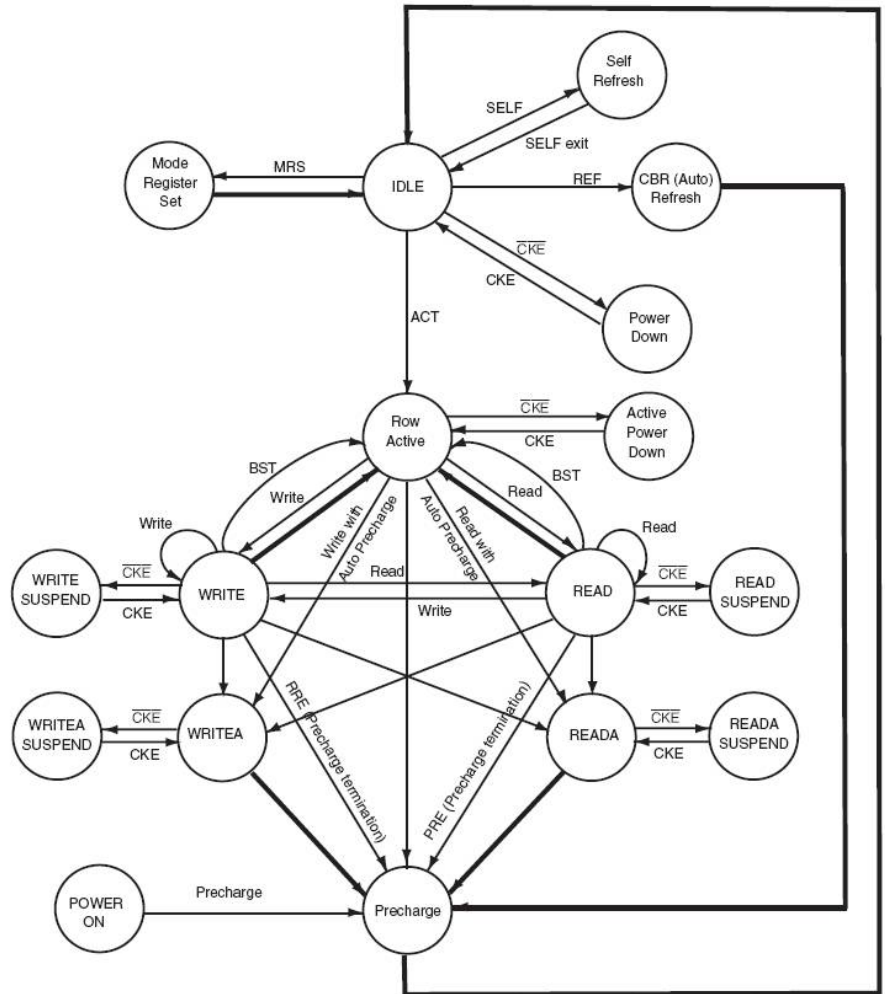
Архитектура ЭВМ

ИУ6

50

# IS42S16320B

## STATE DIAGRAM

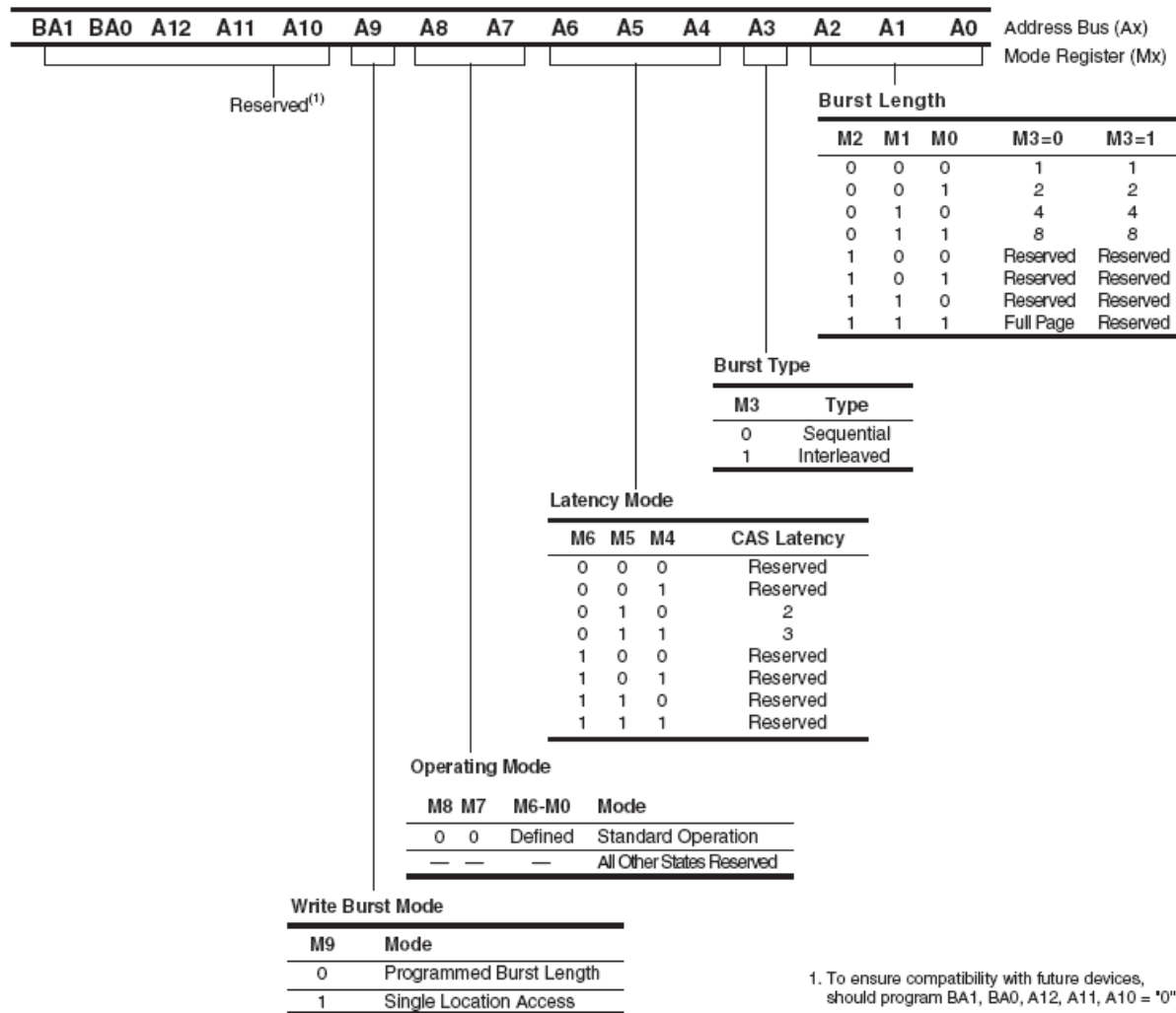


Automatic sequence  
 Manual Input

ПРИМЕР

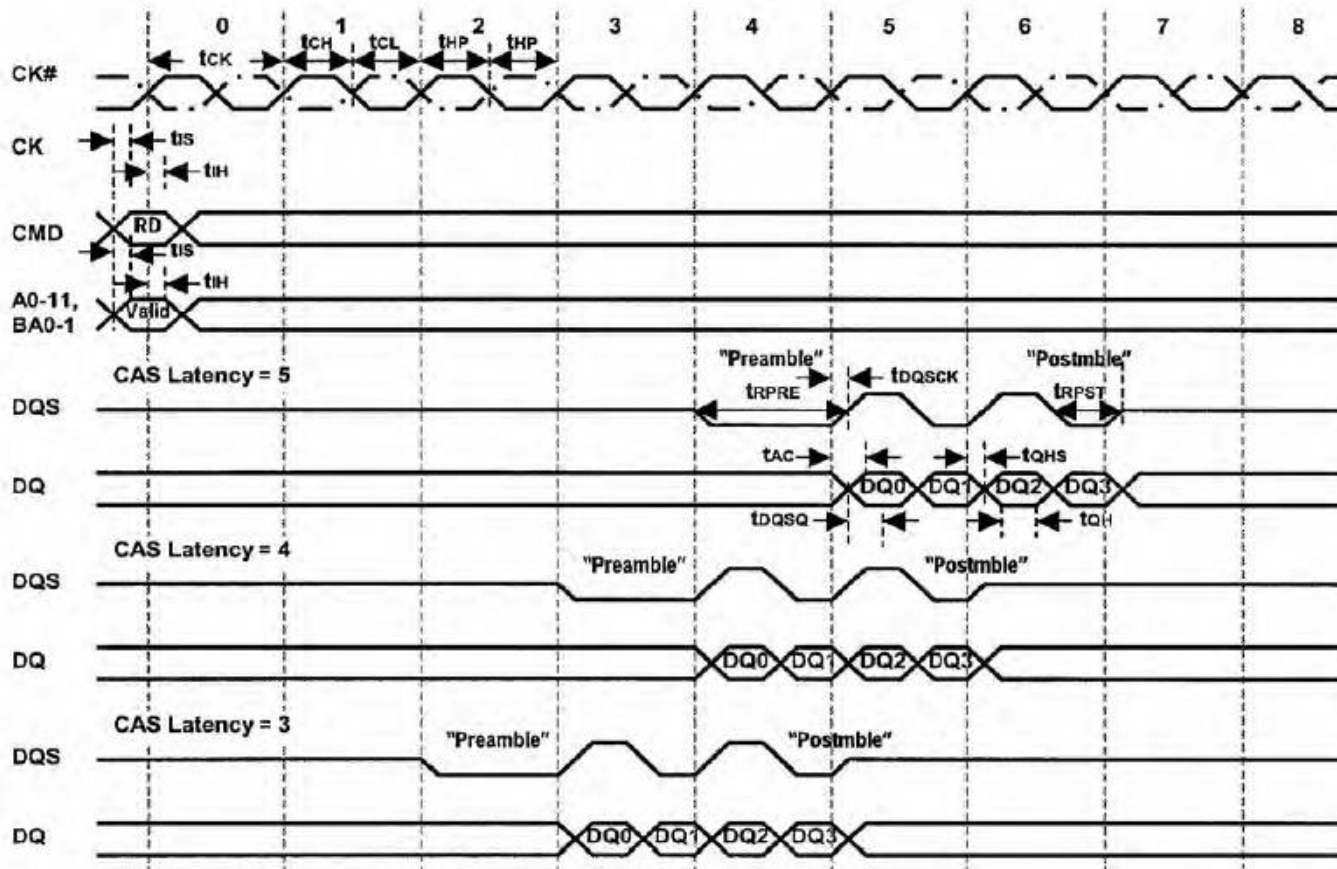
# IS42S16320B

ПРИМЕР



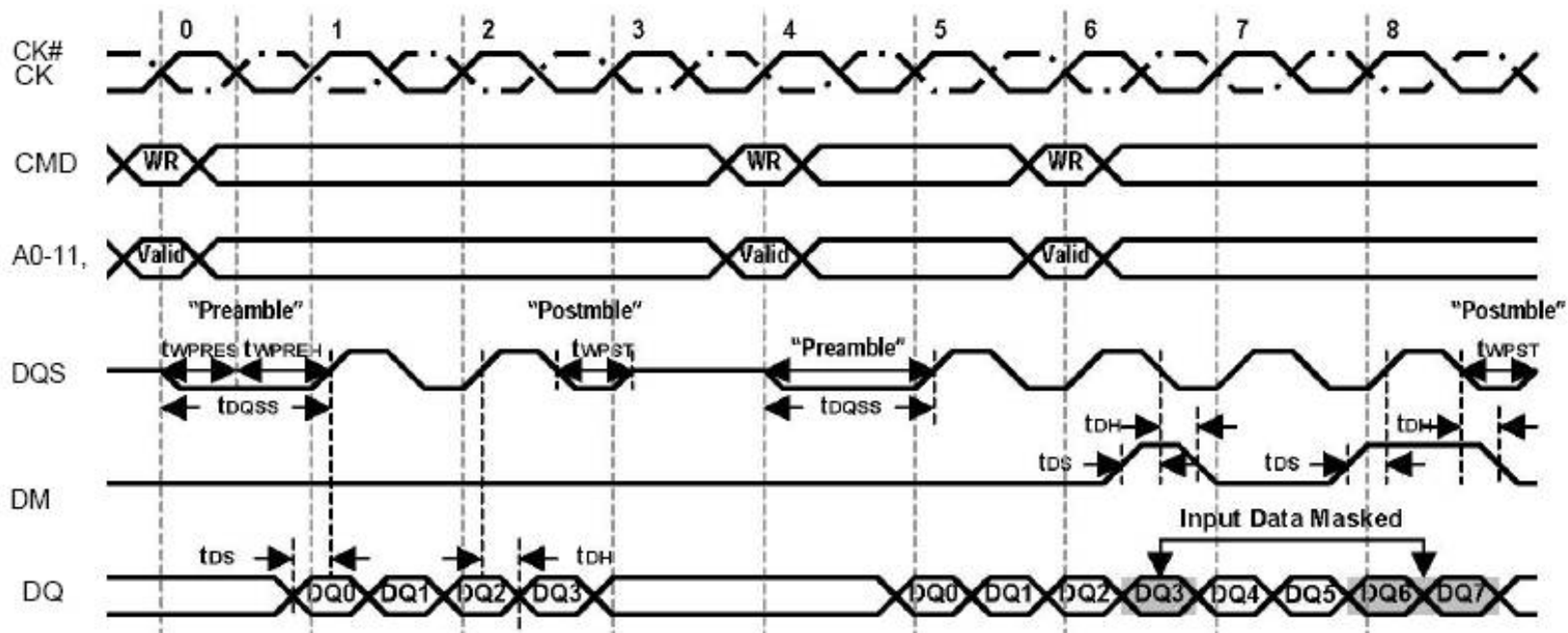


Timing Waveforms  
Figure 1. AC Parameters for Read Timing ( Burst Length =4)



ПРИМЕР

Figure 2. AC Parameters for Write Timing (Burst Length=4 )



ПРИМЕР

# Постоянные запоминающие устройства

МПЗУ (MROM)

ППЗУ (PROM)

РПЗУ-УФ (EPROM)

ОПРПЗУ-УФ (EPROM-ОТР)

РПЗУ-ЭС (EEPROM)

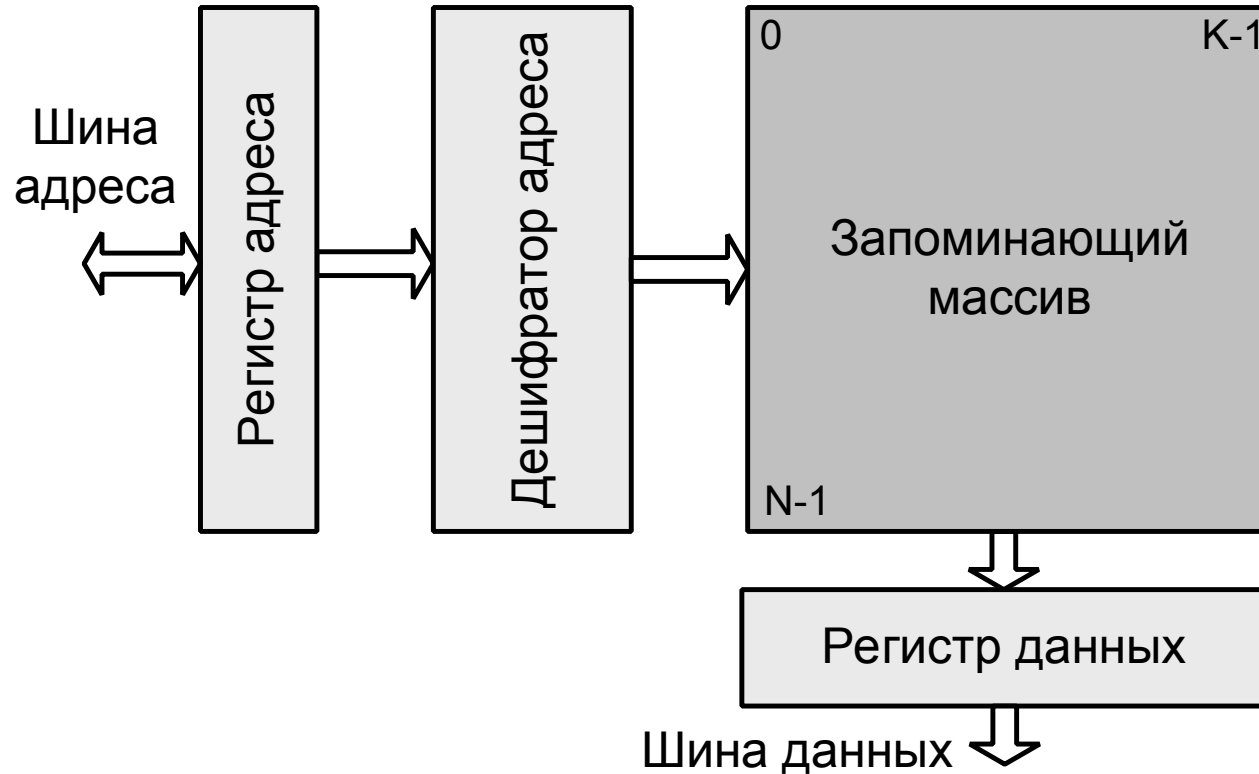
FLASH

Преимущества ROM по сравнению RAM:

- Аппаратная простота.
- Высокая плотность размещения ЗЭ.
- Энергонезависимость.
- Большое быстродействие.

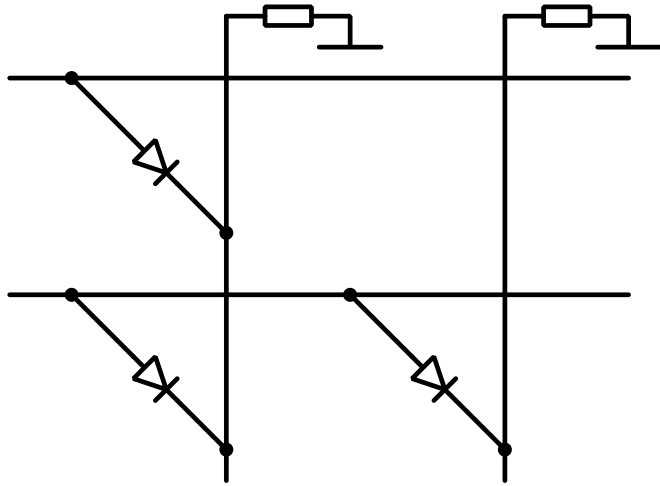


# Структура ПЗУ (ROM)

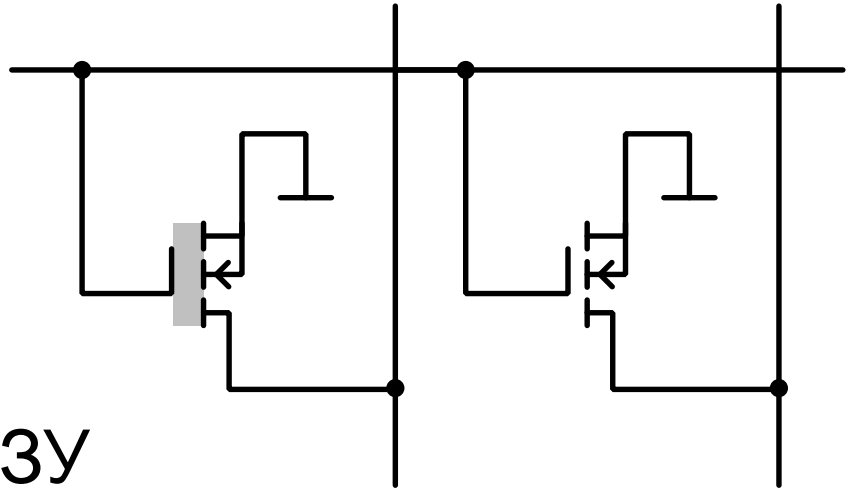


# МПЗУ

## ЗЭ на диодах

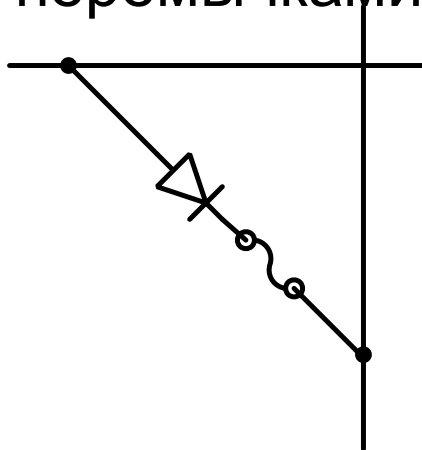


## ЗЭ на МОП транзисторах

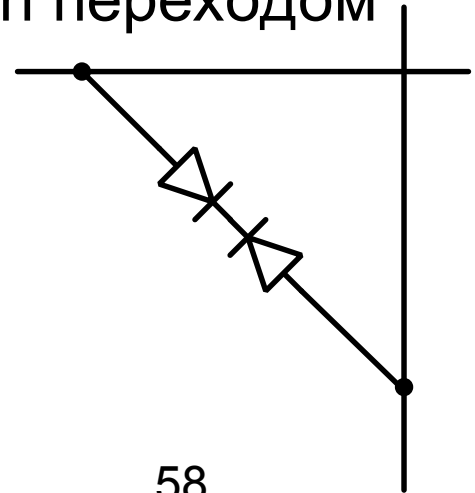


# ППЗУ

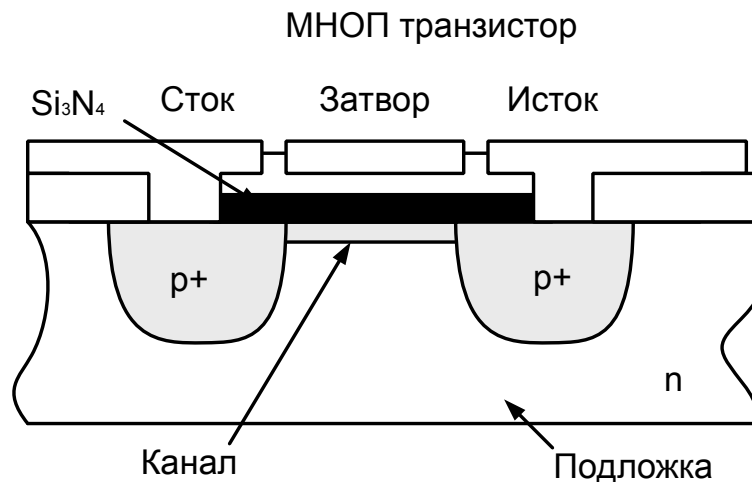
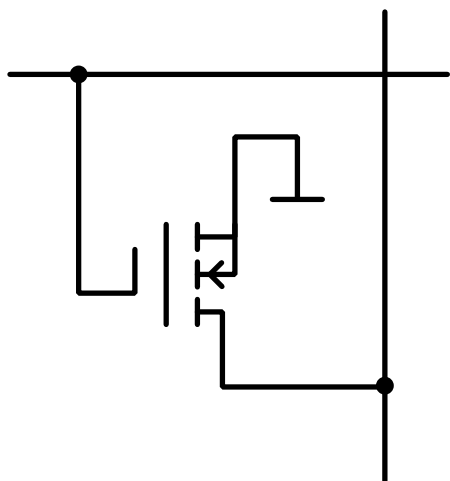
## ППЗУ с плавкими перемычками



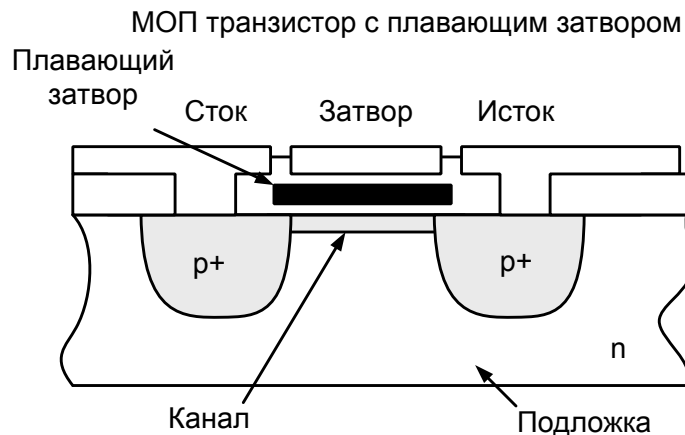
## ППЗУ с пережигаемым р-п переходом



# РПЗУ-УФ, ОПРРПЗУ-УФ (EPROM, EPROM-OTP)



# РПЗУ-ЭС (EEPROM), FLASH





# IS93C76A IS93C86A

## 8K-BIT/16K-BIT SERIAL ELECTRICALLY ERASABLE PROM

MAY 2007

### FEATURES

- Industry-standard Microwire Interface
  - Non-volatile data storage
  - Wide voltage operation:  
V<sub>CC</sub> = 1.8V to 5.5V
  - Auto increment for efficient data dump
- User Configured Memory Organization
  - By 16-bit or by 8-bit
- Hardware and software write protection
  - Defaults to write-disabled state at power-up
  - Software instructions for write-enable/disable
- Enhanced low voltage CMOS E<sup>2</sup>PROM technology
- Versatile, easy-to-use Interface
  - Self-timed programming cycle
  - Automatic erase-before-write
  - Programming status indicator
  - Word and chip erasable
  - Chip select enables power savings
- Durable and reliable
  - 40-year data retention after 1M write cycles
  - 1 million write cycles
  - Unlimited read cycles
  - Schmitt-trigger Inputs
- Industrial and Automotive Temperature Grade
- Lead-free available

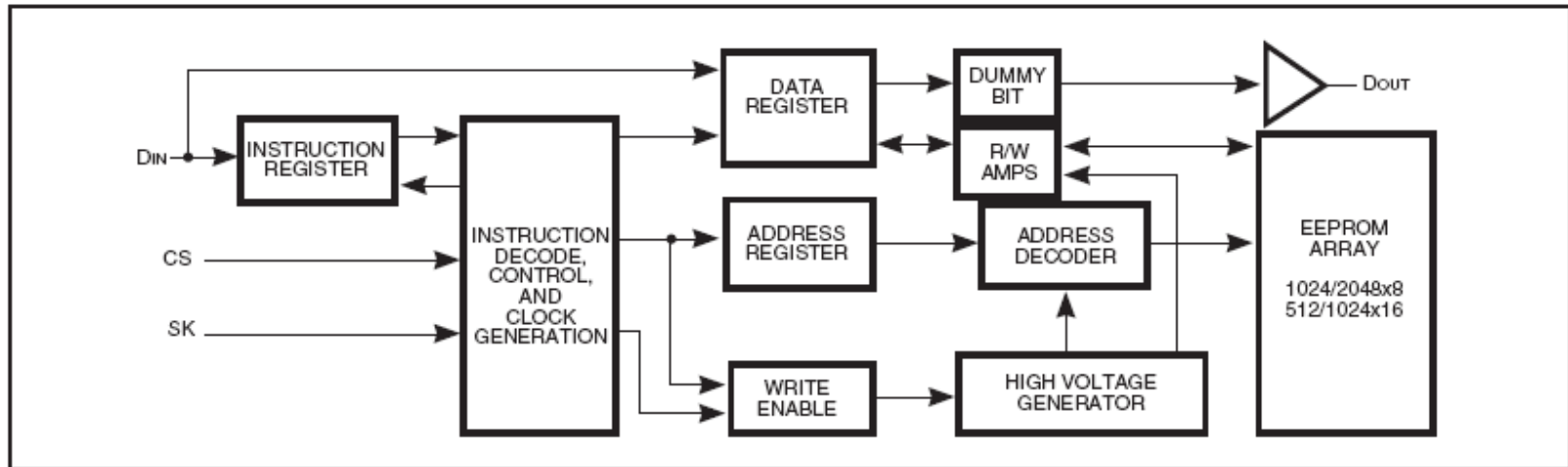


# IS93C76A IS93C86A

## 8K-BIT/16K-BIT SERIAL ELECTRICALLY ERASABLE PROM

MAY 2007

### FUNCTIONAL BLOCK DIAGRAM





# IS93C76A IS93C86A

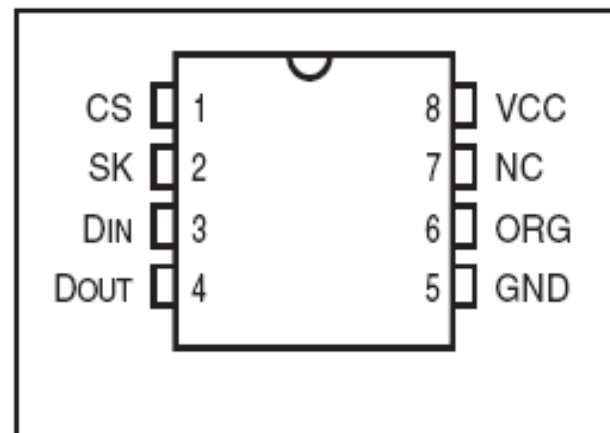
## 8K-BIT/16K-BIT SERIAL ELECTRICALLY ERASABLE PROM

MAY 2007

### PIN DESCRIPTIONS

CS	Chip Select
SK	Serial Data Clock
DIN	Serial Data Input
Dout	Serial Data Output
ORG	Organization Select
NC	Not Connected
Vcc	Power
GND	Ground

### 8-Pin DIP, 8-Pin TSSOP



ПРИМЕР



# IS93C76A IS93C86A

## 8K-BIT/16K-BIT SERIAL ELECTRICALLY ERASABLE PROM

MAY 2007

### INSTRUCTION SET - IS93C86A (16kb)

Instruction <sup>(2)</sup>	Start Bit	OP Code	8-bit Organization (ORG = GND)		16-bit Organization (ORG = Vcc)	
			Address <sup>(1)</sup>	Input Data	Address <sup>(1)</sup>	Input Data
READ	1	10	(A10-A0)	—	(A9-A0)	—
WEN (Write Enable)	1	00	11x xxxx xxxx	—	11 xxxx xxxx	—
WRITE	1	01	(A10-A0)	(D7-D0)	(A9-A0)	(D15-D0)
WRALL (Write All Registers)	1	00	01x xxxx xxxx	(D7-D0)	01 xxxx xxxx	(D15-D0)
WDS (Write Disable)	1	00	00x xxxx xxxx	—	00 xxxx xxxx	—
ERASE	1	11	(A10-A0)	—	(A9-A0)	—
ERAL (Erase All Registers)	1	00	10x xxxx xxxx	—	10 xxxx xxxx	—

#### Notes:

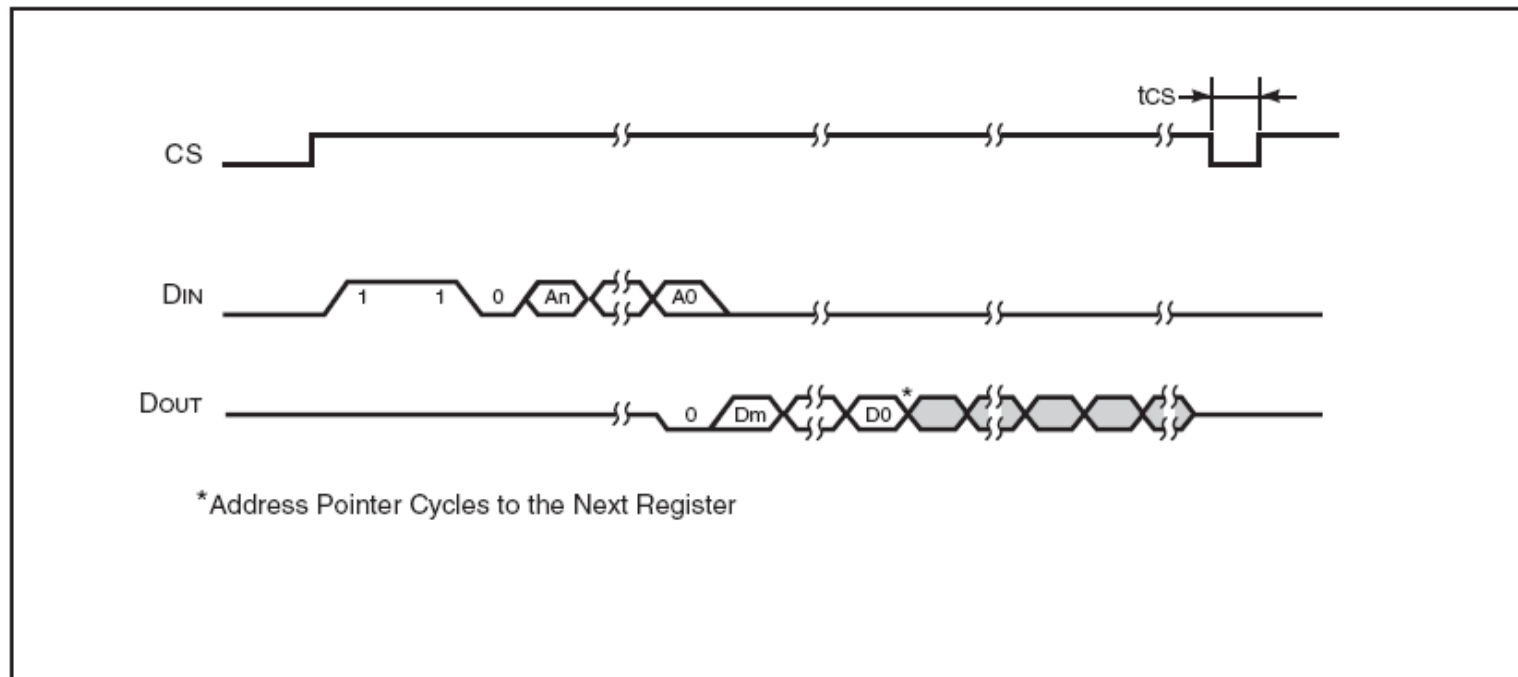
1. x = Don't care bit.
2. If the number of bits clocked-in does not match the number corresponding to a selected command, all extra trailing bits are ignored, and WRITE, WRALL, ERASE, ERAL, WEN, and WDS instructions are rejected, but READ is accepted.

# IS93C76A IS93C86A

## 8K-BIT/16K-BIT SERIAL ELECTRICALLY ERASABLE PROM

MAY 2007

FIGURE 3. READ CYCLE TIMING



ПРИМЕР





# **IS93C76A IS93C86A**

---

**8K-BIT/16K-BIT SERIAL ELECTRICALLY  
ERASABLE PROM**

MAY 2007

ПРИМЕР



---

**2-Megabit  
5-volt Only  
Serial  
DataFlash<sup>®</sup>**

---

**AT45D021**

**Pin Configurations**

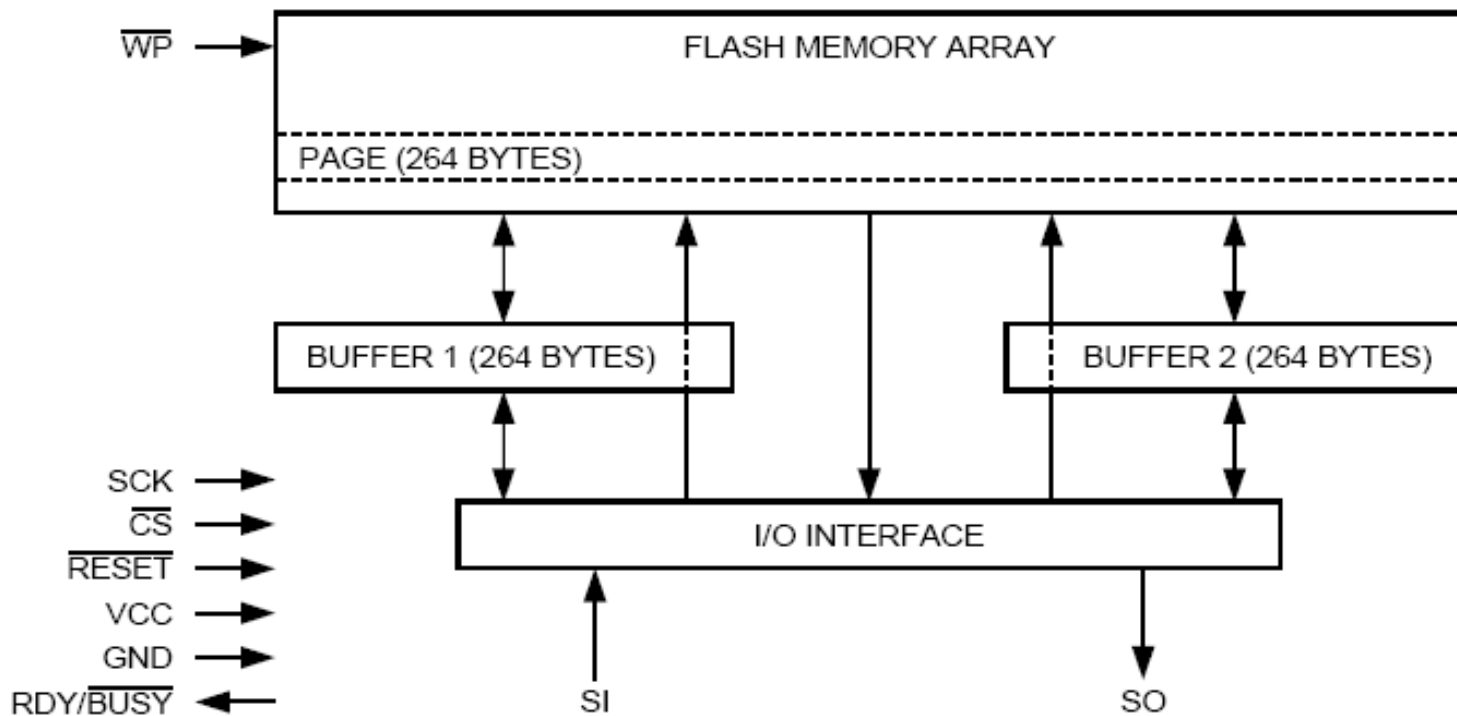
Pin Name	Function
CS	Chip Select
SCK	Serial Clock
SI	Serial Input
SO	Serial Output
WP	Hardware Page Write Protect Pin
$\overline{\text{RESET}}$	Chip Reset
$\text{RDY}/\overline{\text{BUSY}}$	Ready/Busy

**Features**

- Single 4.5V - 5.5V Supply
- Serial Interface Architecture
- Page Program Operation
  - Single Cycle Reprogram (Erase and Program)
  - 1024 Pages (264 Bytes/Page) Main Memory
- Two 264-Byte SRAM Data Buffers – Allows Receiving of Data while Reprogramming of Nonvolatile Memory
- Internal Program and Control Timer
- Fast Page Program Time – 7 ms Typical
- 80  $\mu\text{s}$  Typical Page to Buffer Transfer Time
- Low Power Dissipation
  - 15 mA Active Read Current Typical
  - 15  $\mu\text{A}$  CMOS Standby Current Typical
- 10 MHz Max Clock Frequency
- Hardware Data Protection Feature
- Serial Peripheral Interface (SPI) Compatible – Modes 0 and 3
- CMOS and TTL Compatible Inputs and Outputs
- Commercial and Industrial Temperature Ranges



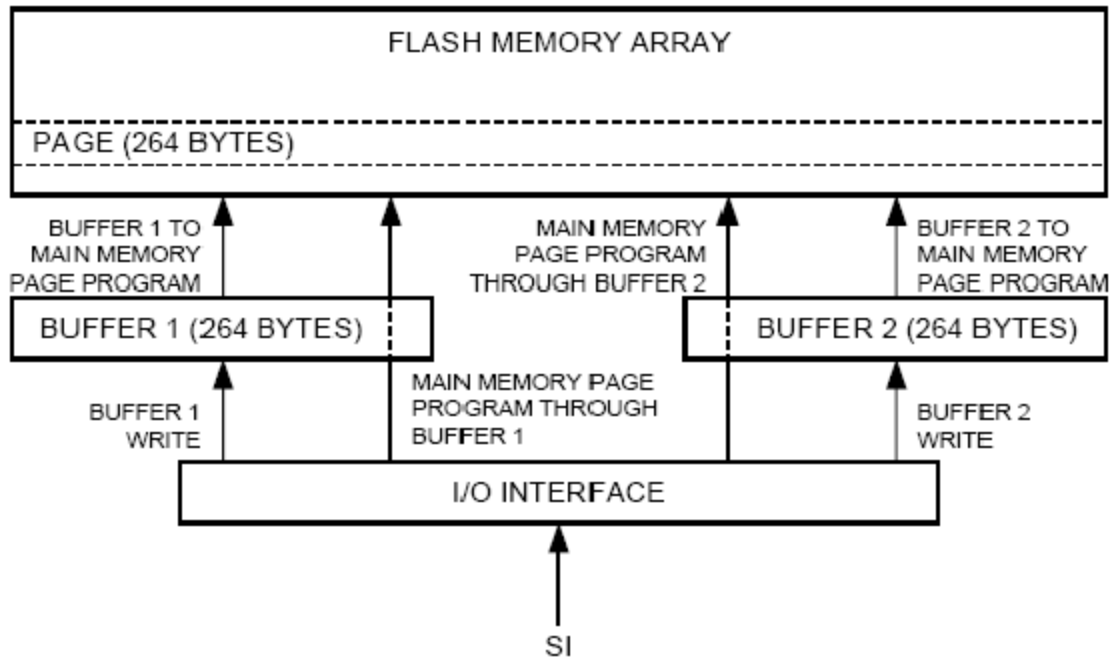
# Block Diagram



ПРИМЕР

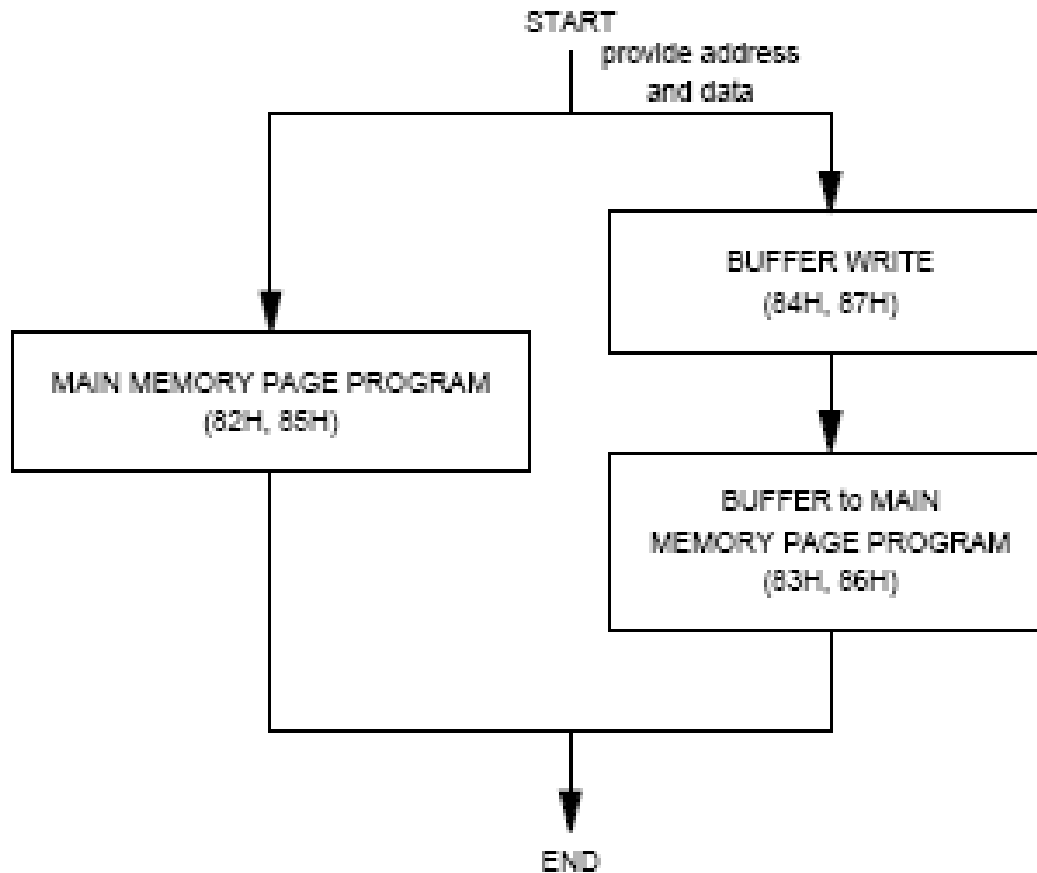
## Write Operations

The following block diagram and waveforms illustrate the various write sequences available.



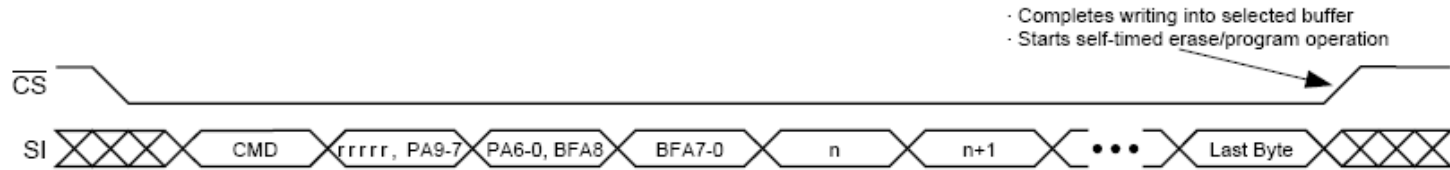
ПРИМЕР

## Algorithm for Programming or Reprogramming of the Entire Array Sequentially



ПРИМЕР

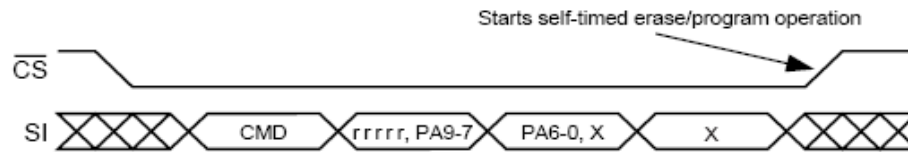
## Main Memory Page Program through Buffers



## Buffer Write



## Buffer to Main Memory Page Program (Data from Buffer Programmed into Flash Page)

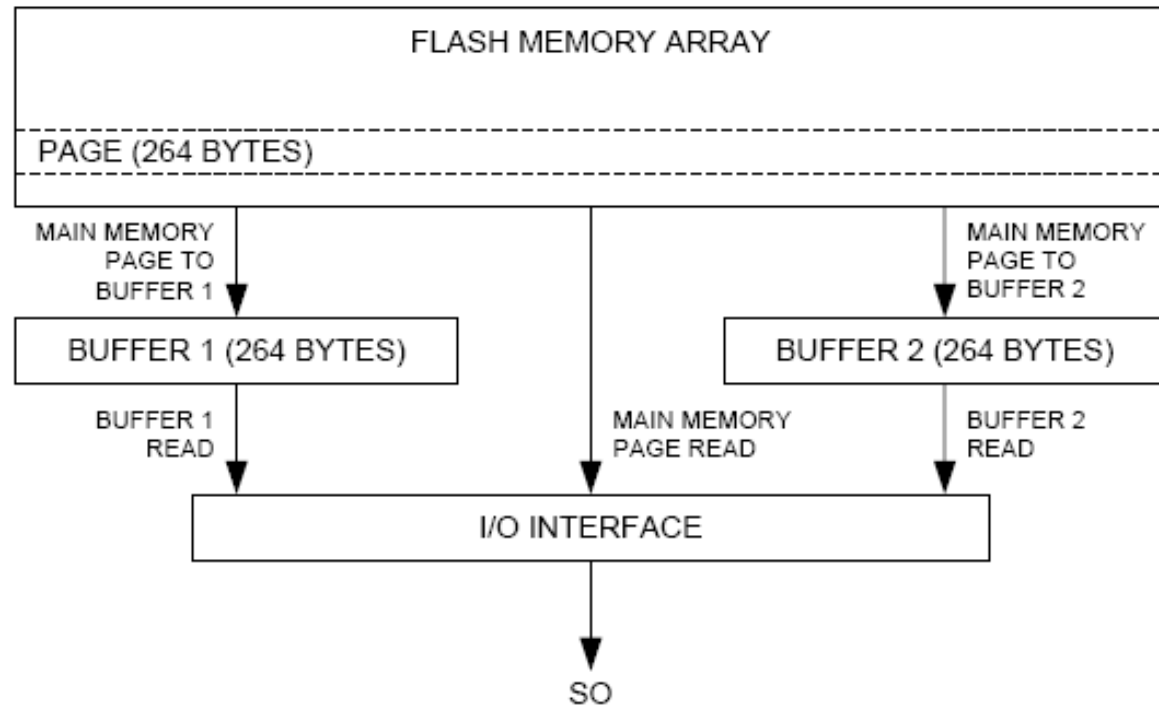


Each transition represents  
8 bits and 8 clock cycles

n = 1st byte read  
n+1 = 2nd byte read

## Read Operations

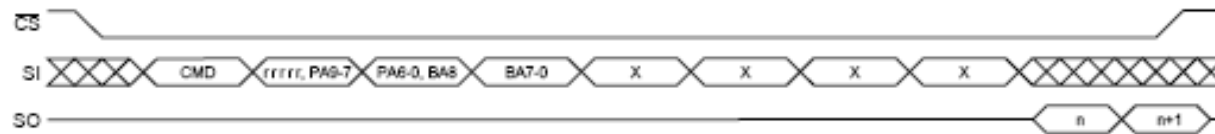
The following block diagram and waveforms illustrate the various read sequences available.



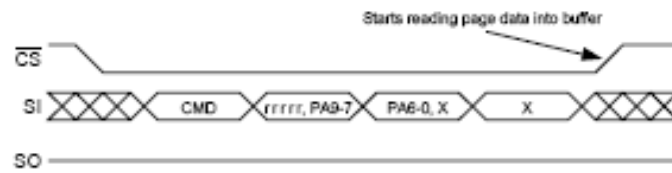
ПРИМЕР



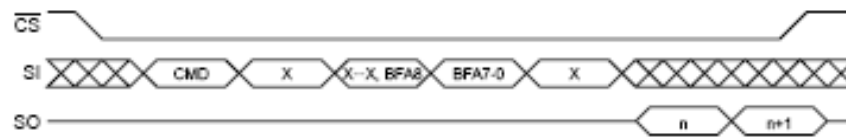
### Main Memory Page Read



### Main Memory Page to Buffer Transfer (Data from Flash Page Read into Buffer)



### Buffer Read



Each transition represents 8 bits and 8 clock cycles

n = 1st byte written  
n+1 = 2nd byte written



# Методы повышение надежности ЗУ

## Контроль по четности/нечетности

$$P_ч = d_0 \oplus d_1 \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_5 \oplus d_6 \oplus d_7, P_н = \neg P_ч$$

$\oplus$  - операция сложения по модулю

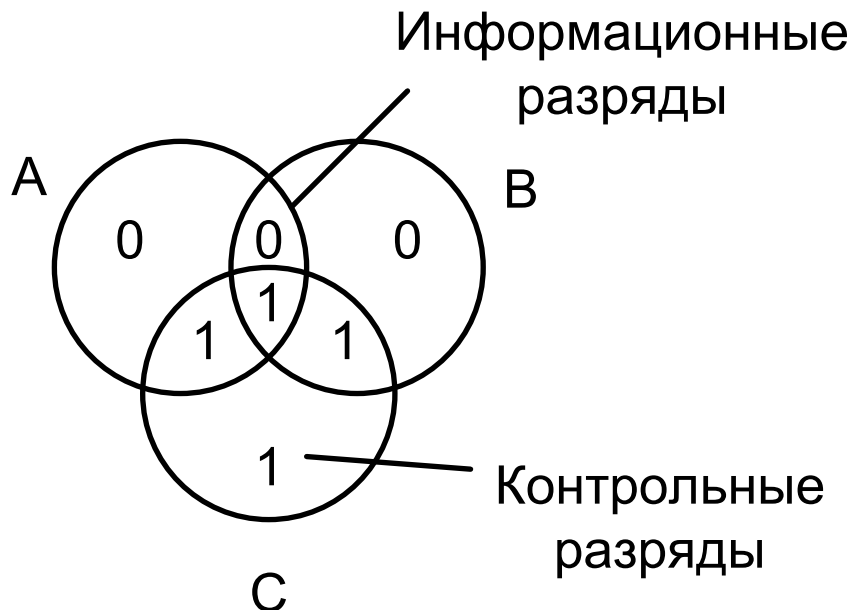
Пример:  $D = 10010100$ , количество единиц = 3,

$$P_ч = d_0 \oplus d_1 \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_5 \oplus d_6 \oplus d_7 = 1, P_н = \neg P_ч = 0$$

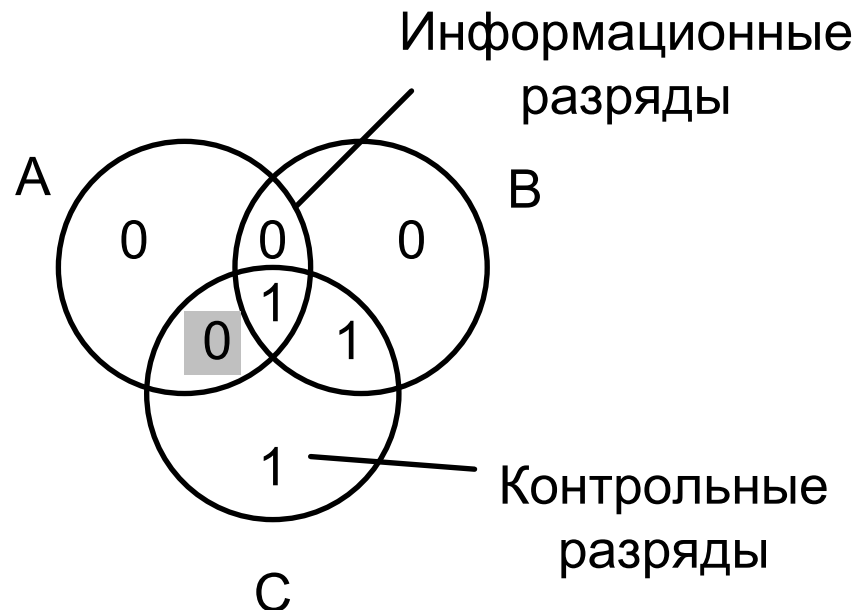
При чтении новое  $P'$  сравнивается  $P$  и если  $P' \oplus P = 1$ , то обнаружена ошибка.

# Код Хэмминга

Исходные данные



Ошибочные данные



$P'_A=1, P'_B=0, P'_C=0 \Rightarrow$  Нарушен информационный бит  $A \cap C / B$

Результат проверок по коду Хэмминга - синдром:

$$S = \{p_1 \oplus p'_1, p_2 \oplus p'_2, p_3 \oplus p'_3\}$$

Код Хэмминга позволяет обнаружить и исправить единичную ошибку и обнаружить двойную.

-Если  $S = 0$ , то ошибок не обнаружено.

-Если в синдроме одна единица, то ошибка в одном корректирующем разряде (не исправляется).

-Если в синдроме несколько единиц, то он указывает на ошибочный информационный разряд.

-При добавлении общего контрольного разряда ( $P = d_0 \oplus d_1 \oplus d_2 \oplus d_3 \oplus p_0 \oplus p_1 \oplus p_2$ ) можно обнаружить двойную ошибку (не исправляется)

# Пример для 4-х разрядных информационных слов

Корректирующие разряды размещены в позициях  $2^i$  и контролируют разряды с двоичным номером, содержащим  $2^i$ .

P	d3	d2	d1	p2	d0	p1	p0
---	----	----	----	----	----	----	----

8 7 6 5 4 3 2 1

Исходное слово

0	0	1	0	1	1	0	1
---	---	---	---	---	---	---	---

8 7 6 5 4 3 2 1

Ошибочное слово

0	0	1	0	1	0	0	1
---	---	---	---	---	---	---	---

8 7 6 5 4 3 2 1

Синдром:  $S = \{p_2 \oplus p'_2, p_1 \oplus p'_1, p_0 \oplus p'_0\} = 011_2 = 3_{10}$

$$p_0 = d_0 \oplus d_1 \oplus d_3,$$

$$p_1 = d_0 \oplus d_2 \oplus d_3,$$

$$p_2 = d_1 \oplus d_2 \oplus d_3$$

$$p_0 = d_0 \oplus d_1 \oplus d_3 = 1,$$

$$p_1 = d_0 \oplus d_2 \oplus d_3 = 0,$$

$$p_2 = d_1 \oplus d_2 \oplus d_3 = 1$$

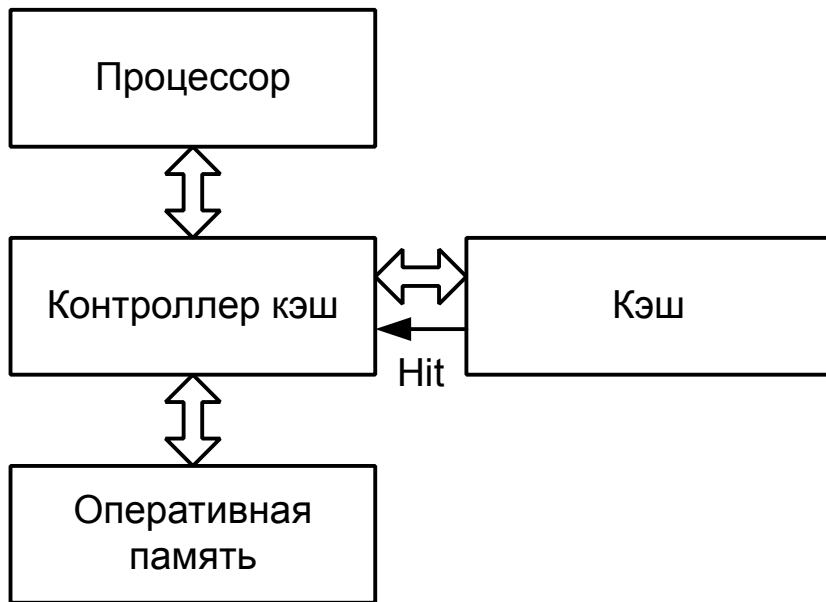
$$p'_0 = d_0 \oplus d_1 \oplus d_3 = 0,$$

$$p'_1 = d_0 \oplus d_2 \oplus d_3 = 1,$$

$$p'_2 = d_1 \oplus d_2 \oplus d_3 = 1$$

# Принципы построения кэш-памяти

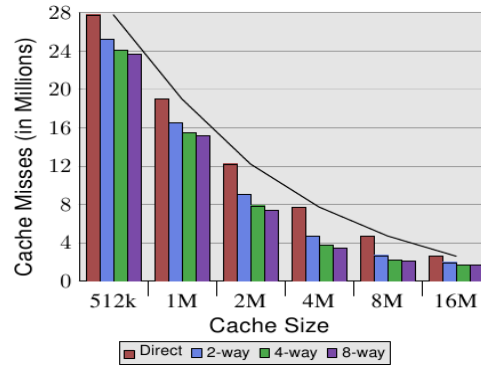
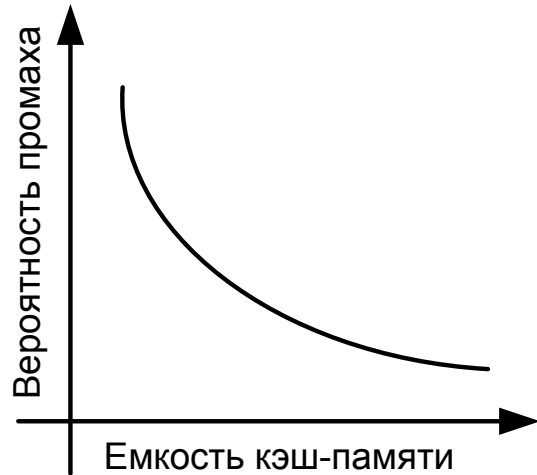
Кэш-память – ассоциативное ЗУ, позволяющее сгладить разрыв в производительности процессора и оперативной памяти. Выборка из кэш-памяти осуществляется по физическому адресу ОП.



Эффективность кэш-памяти зависит от:

- Емкости кэш-памяти.
- Размера строки.
- Способа отображения ОП в кэш.
- Алгоритма замещения информации в кэш.
- Алгоритма согласования ОП и кэш.
- Числа уровней кэш.

## Ёмкость кэш-памяти



## Размер линейки



Способы отображения ОП в кэш:

- Произвольная загрузка.
- Прямое размещение.
- Наборно-ассоциативный способ отображения.

# Произвольная загрузка (Fully associated cache memory, FACM).

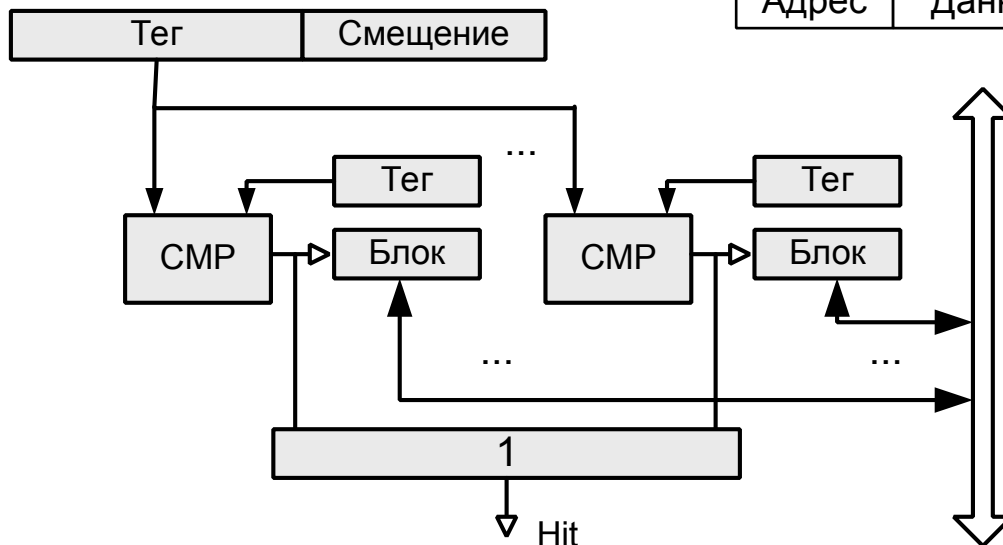
Адрес строки FACM определяется из условия формирования наиболее представительной выборки

КЭШ

ОП

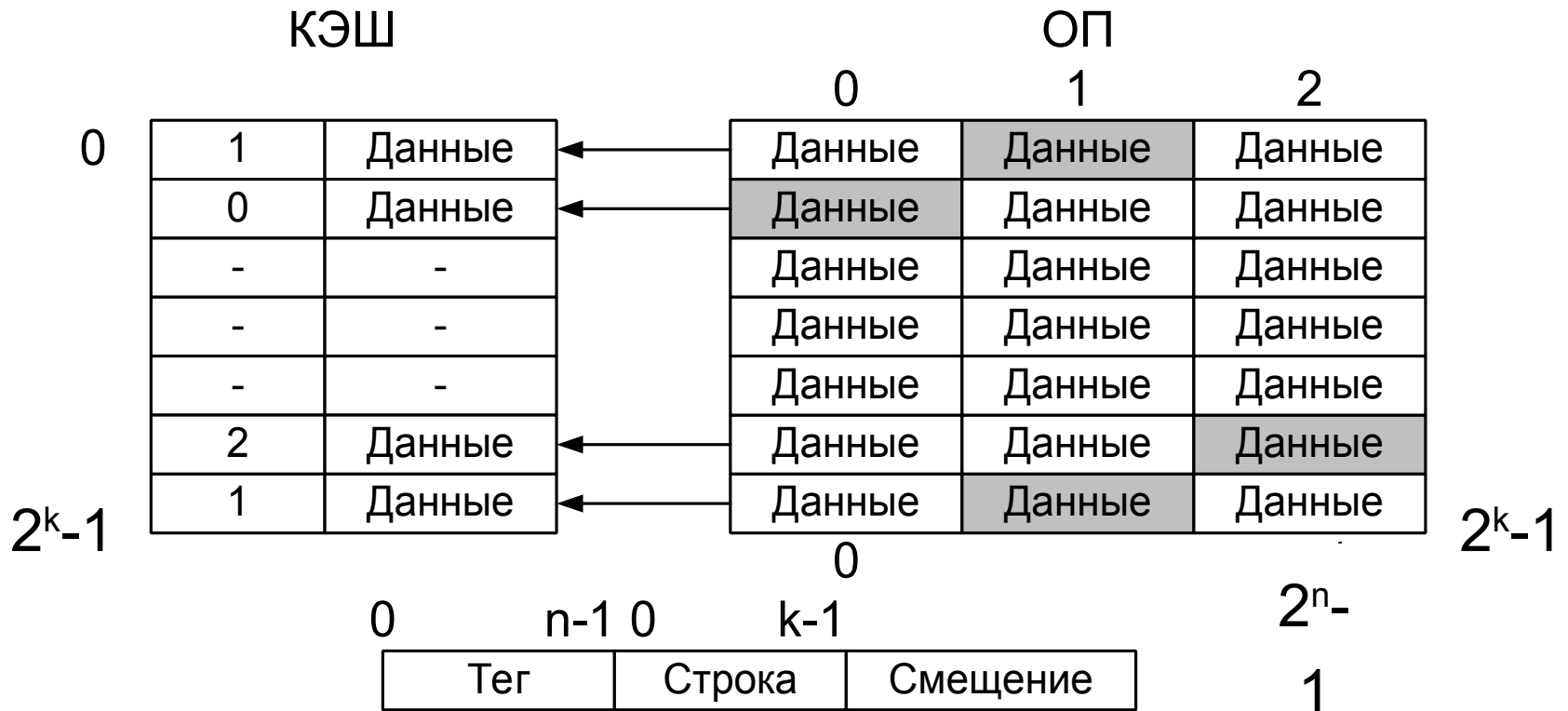
Адрес	Данные
Адрес	Данные
-	-
Адрес	Данные
-	-
Адрес	Данные
Адрес	Данные

Данные
Данные
-
Данные
-
Данные
Данные
-
Данные
Данные
...



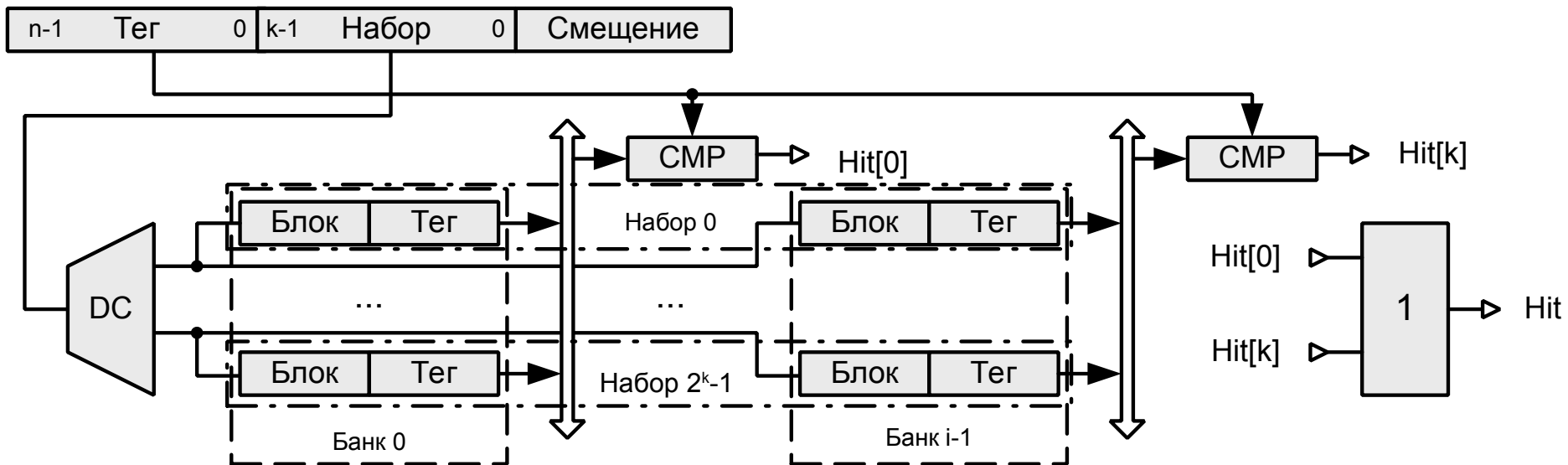
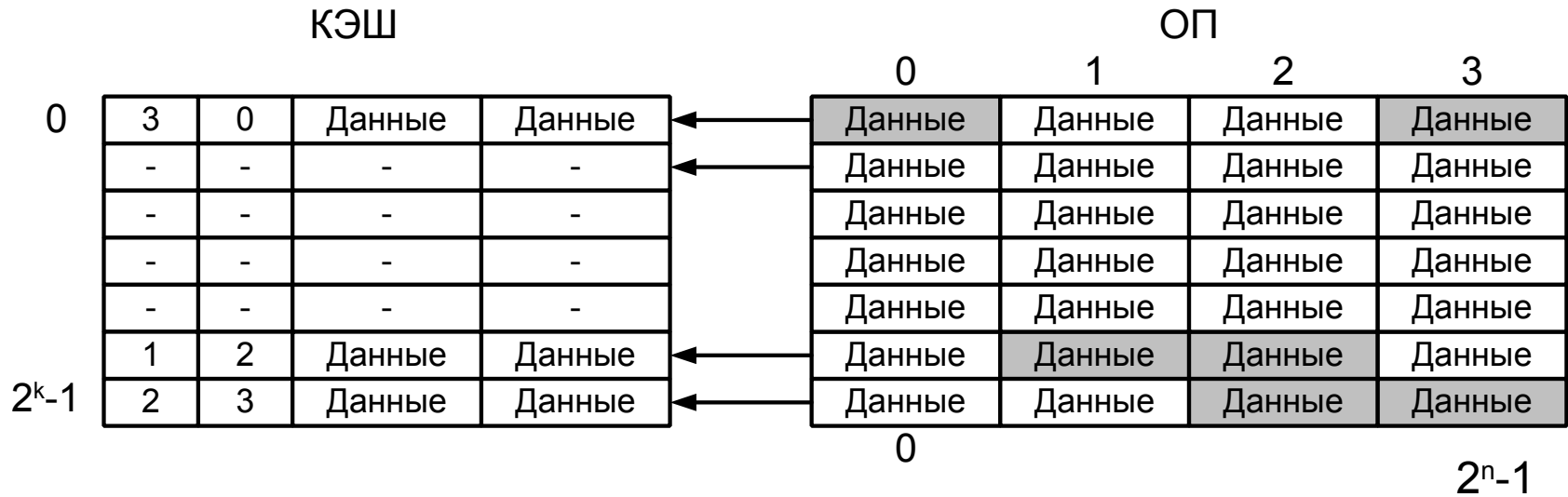
# Прямое размещение.

Адрес строки однозначно определяется по тегу ( $i = t \bmod k$ ).





# Наборно-ассоциативная кэш-память (Set associated cache memory)



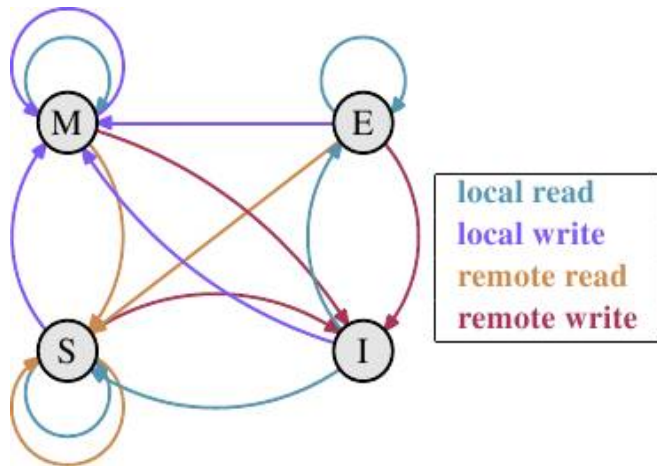
# Алгоритмы замещения

- Замещение немодифицированных данных.
- Рандомизированный алгоритм.
- Замещение наименее используемого (Least Recently Used, LRU)

## Согласование ОП и кэш

- Метод сквозной записи (Write True).
- Метод сквозной записи с буферизацией (Write Combining).
- Метод обратной записи (Write Back).

## Протокол MESI

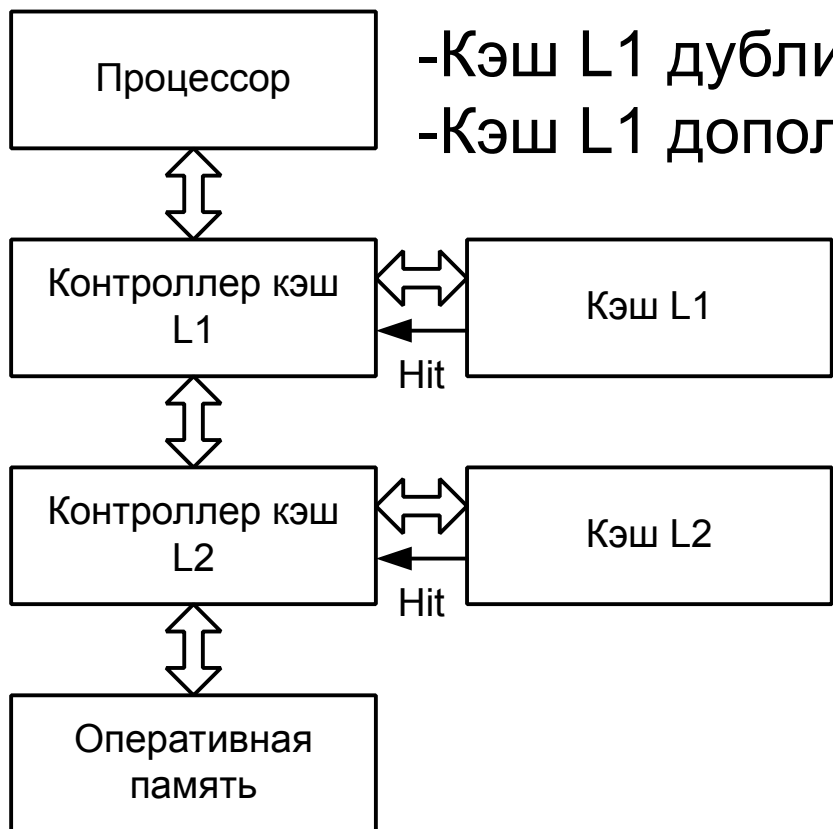


**Modified**  
**Exclusive**  
**Shared**  
**Invalid**

- Признак несогласованных данных.
- Признак согласованных данных.
- Признак согласованных данных в ВС.
- Признак отсутствия данных.

\* - <http://lwn.net/Articles/252125/>

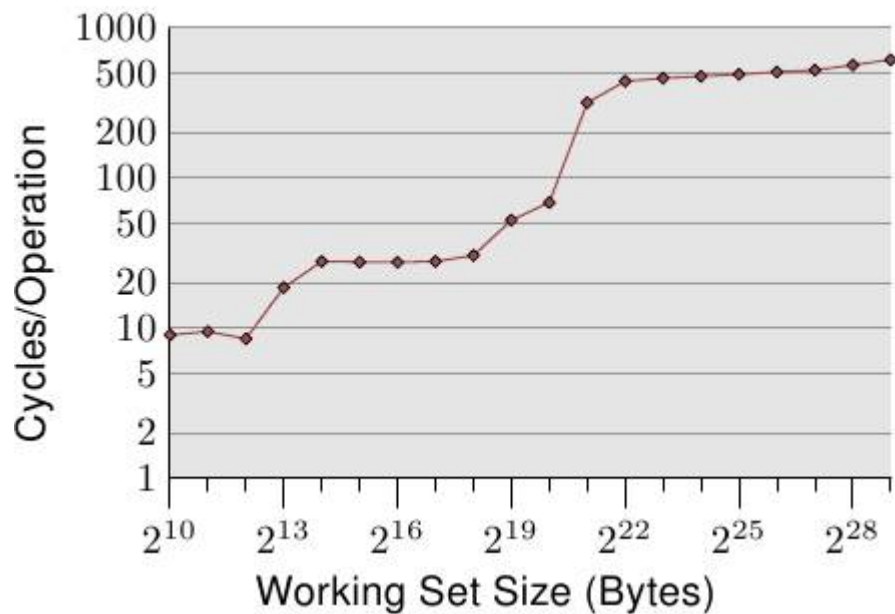
# Разделение кэш-памяти



- Кэш L1 дублирует L2 (inclusive).
- Кэш L1 дополняет L2 (exclusive).

Доступ к массивам данным по случайным адресам

L1D —  $2^{13}$  байт  
L2D —  $2^{21}$  байт



# Виртуальная память

Механизм виртуализации адресного пространства позволяет:

- Увеличить объем адресуемой памяти.
- Использовать физическую память различного объема.
- Возложить на аппаратную составляющую механизмы доступа к ВЗУ
- Сгладить разрыв в производительности ОП и ВЗУ.
- Ускоряет доступ к данным по последовательным адресам.
- Способствует реализации защиты памяти.

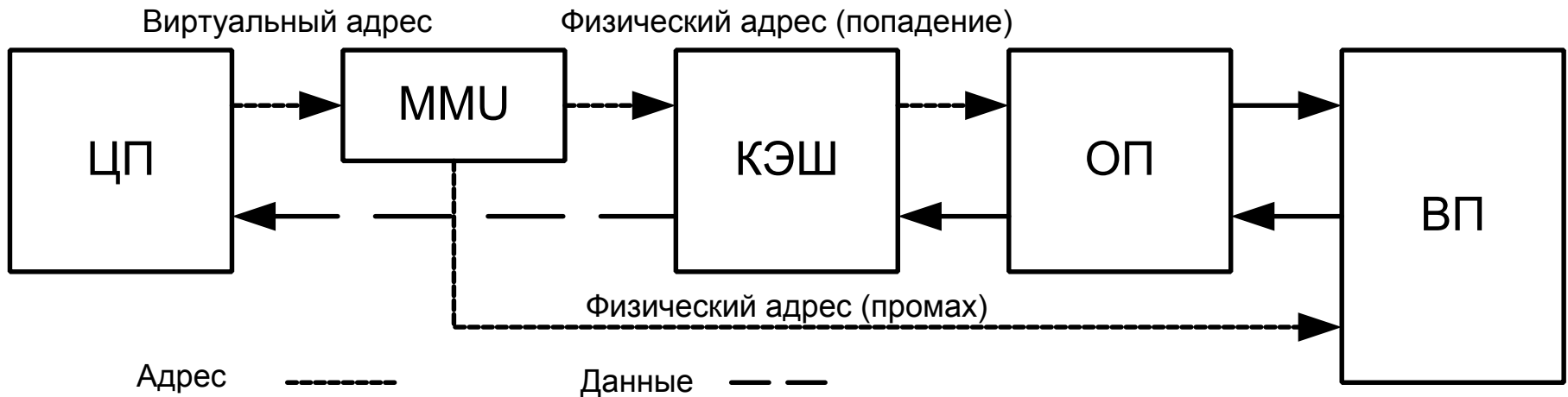
Виртуальные системы строятся по трем принципам:

- Системы с блоками различного размера (сегментная организация).
- Системы с блоками одинакового размера (страничная организация).
- Смешанные системы (сегментно-страничная организация).

# Страничная организация

Программа отображается в память равными блоками – страницами. Преобразование логического адреса в физический осуществляется с помощью таблицы страниц.

Преобразование логического адреса в физический реализуется в устройстве управления памятью (Memory Manage Unit), который определяет, находится ли страница в физической памяти (попадение).



# Схема страничного преобразования

Базовый регистр таблицы страниц

Адрес таблицы страниц

Логический адрес

Номер страницы | Смещение

V - признак присутствия  
страницы в физ. памяти.  
R - признак использования  
страницы.  
M - признак модификации.  
A - признак права  
доступа.

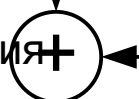
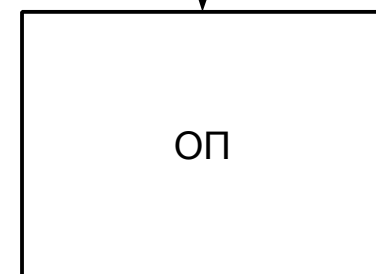
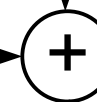


Таблица страниц

V	R	M	A	Номер физ. страницы

V=1



V=0

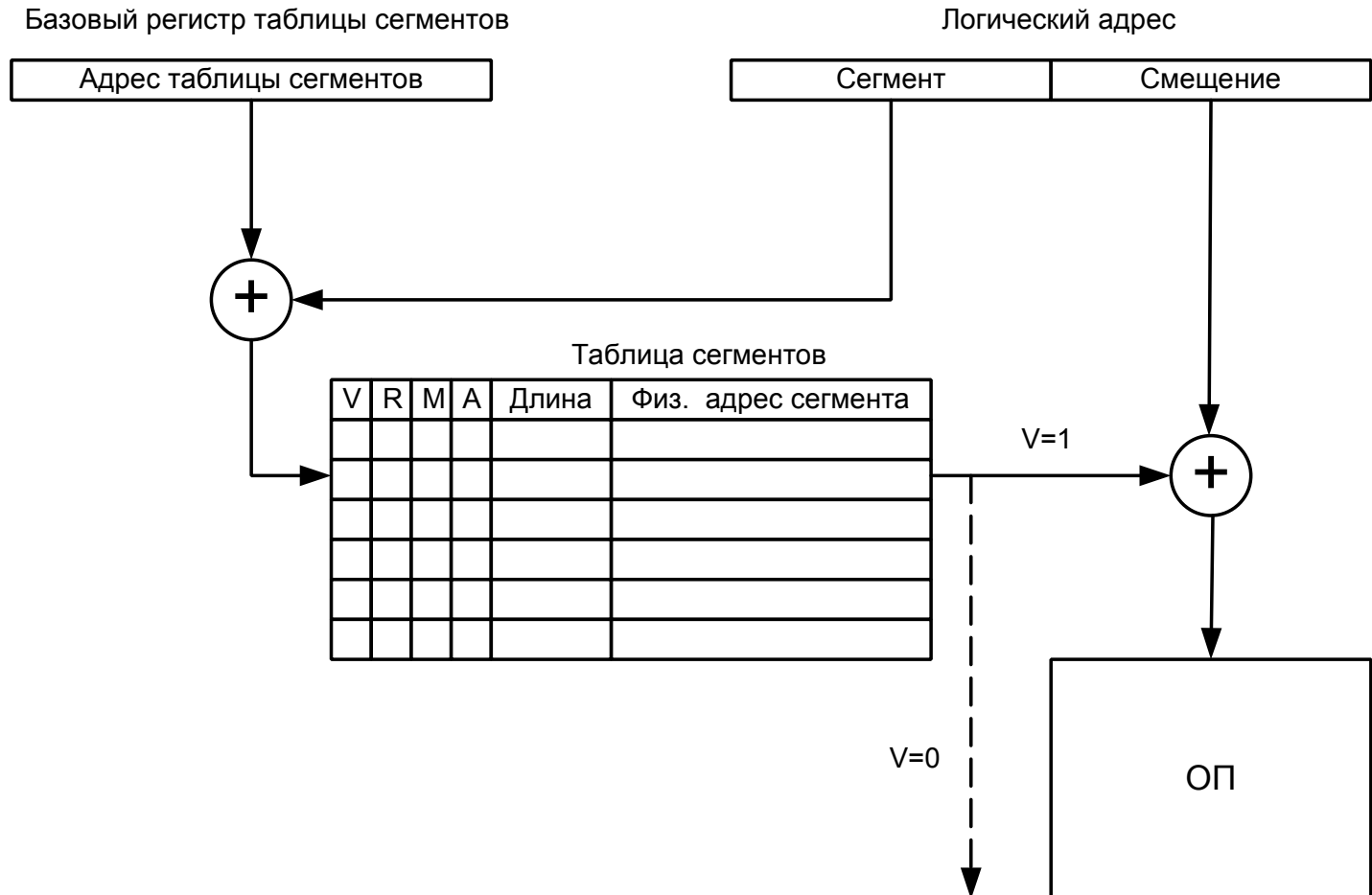
Прерывание

Структура TLB

Номер лог. страницы	V	R	M	A	Номер физ. страницы

# Сегментная организация

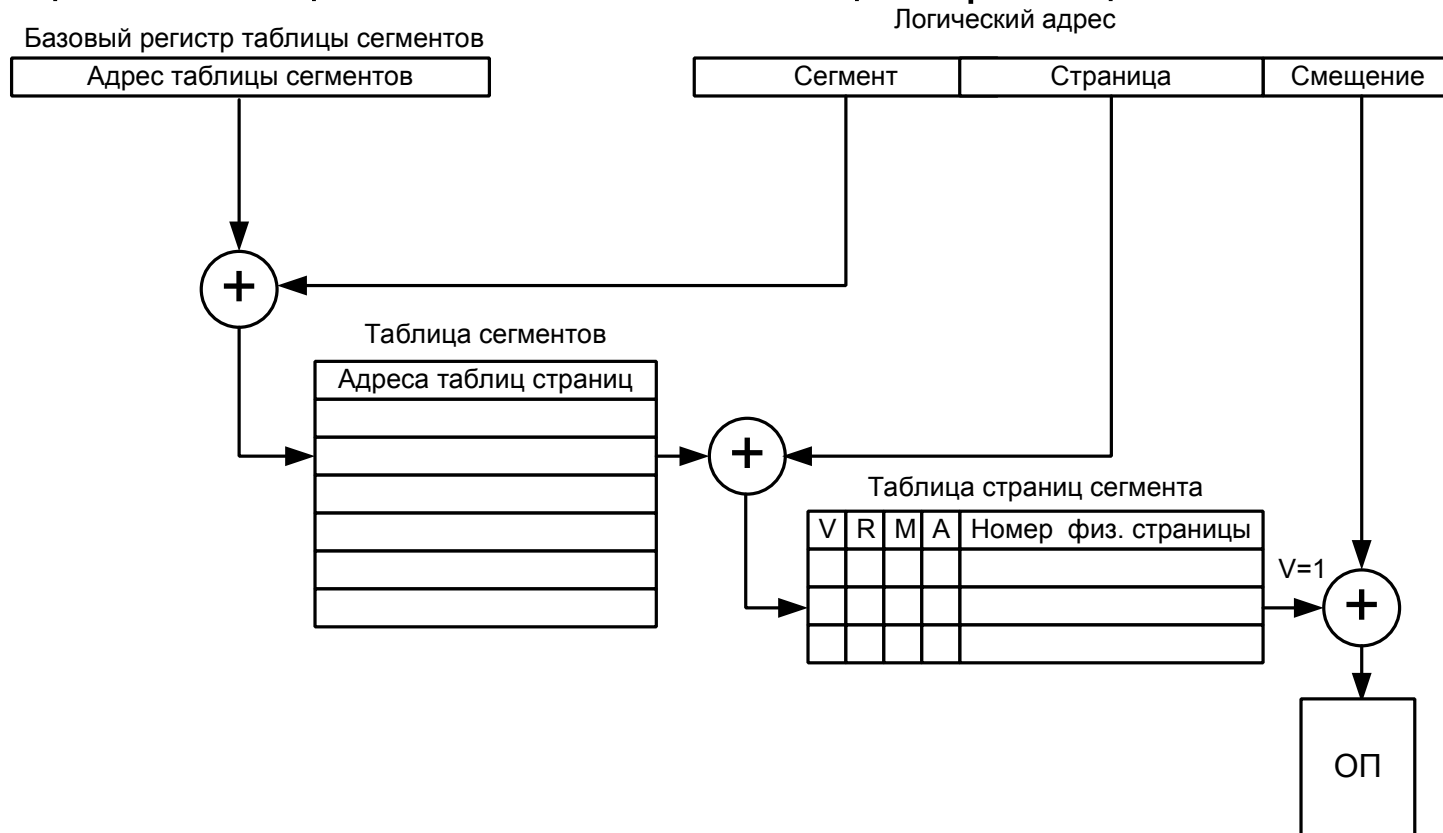
Программа отображается в память блоками различного размера – сегментами. Преобразование логического адреса в физический осуществляется с помощью таблицы сегментов.



# Сегментно-страничная организация памяти

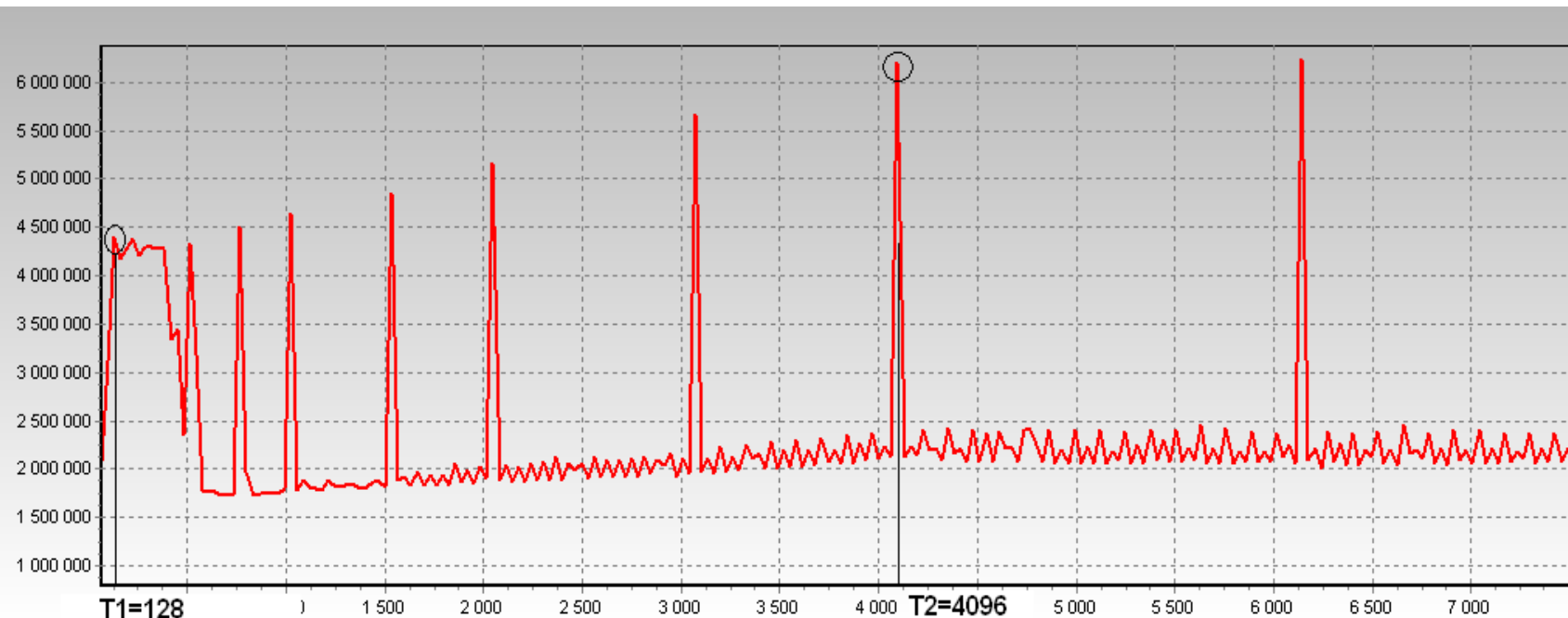
Программа отображается в память блоками различного размера – сегментами, каждый из которых целое число страниц.

Преобразование логического адреса в физический осуществляется с помощью таблицы сегментов и таблицы страниц сегмента.





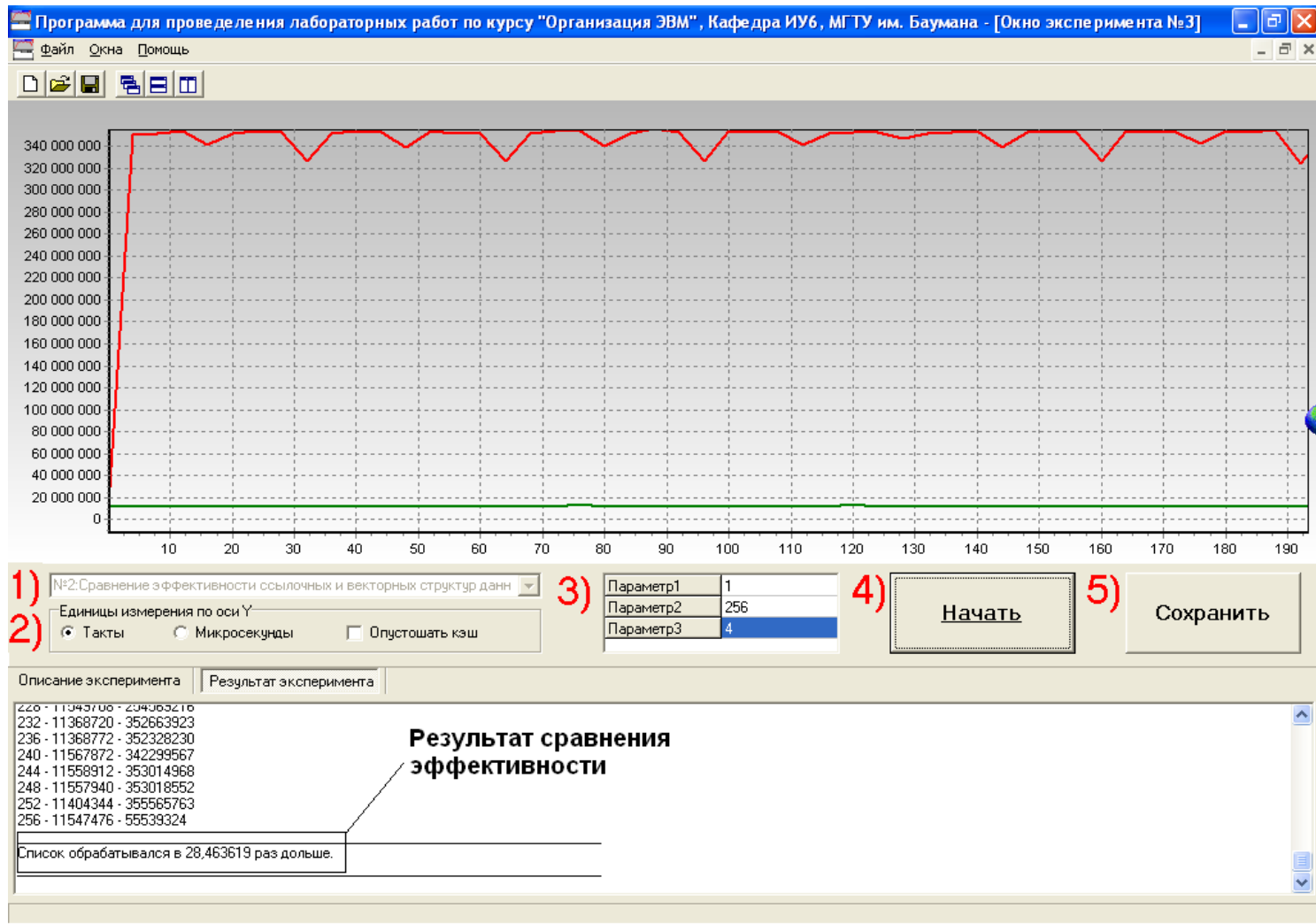
# Исследование расслоения динамической памяти.



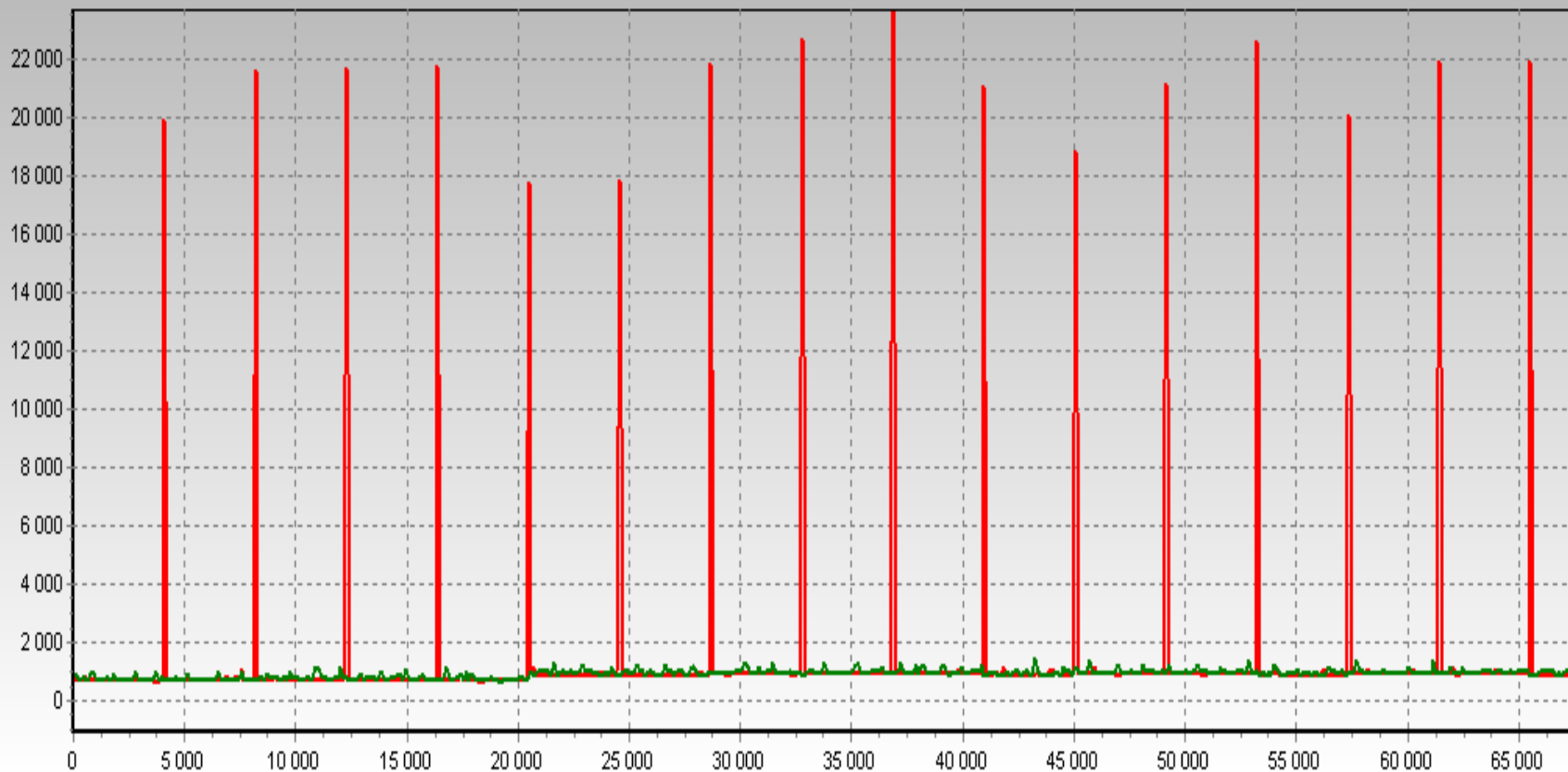
Код профилируемой программы на языке С.

```
// ВЫДЕЛЕНИЕ ПАМЯТИ
p = (int*)_malloc64(Param_[3]); // АДРЕС КРАТЕН 64
for (int pg_size = Param_[2]; pg_size <= Param_[1]; pg_size += Param_[2])
{
    Start_Count(); // Начало замера времени
    volatile int x = 0;
    for (int b = 0; b < pg_size; b += Param_[2])
        for (int a = b; a < Param_[3]; a += pg_size)
            x += *(int *) (int(p) + a);
    Finish_Count(); // Конец замера времени
}
```

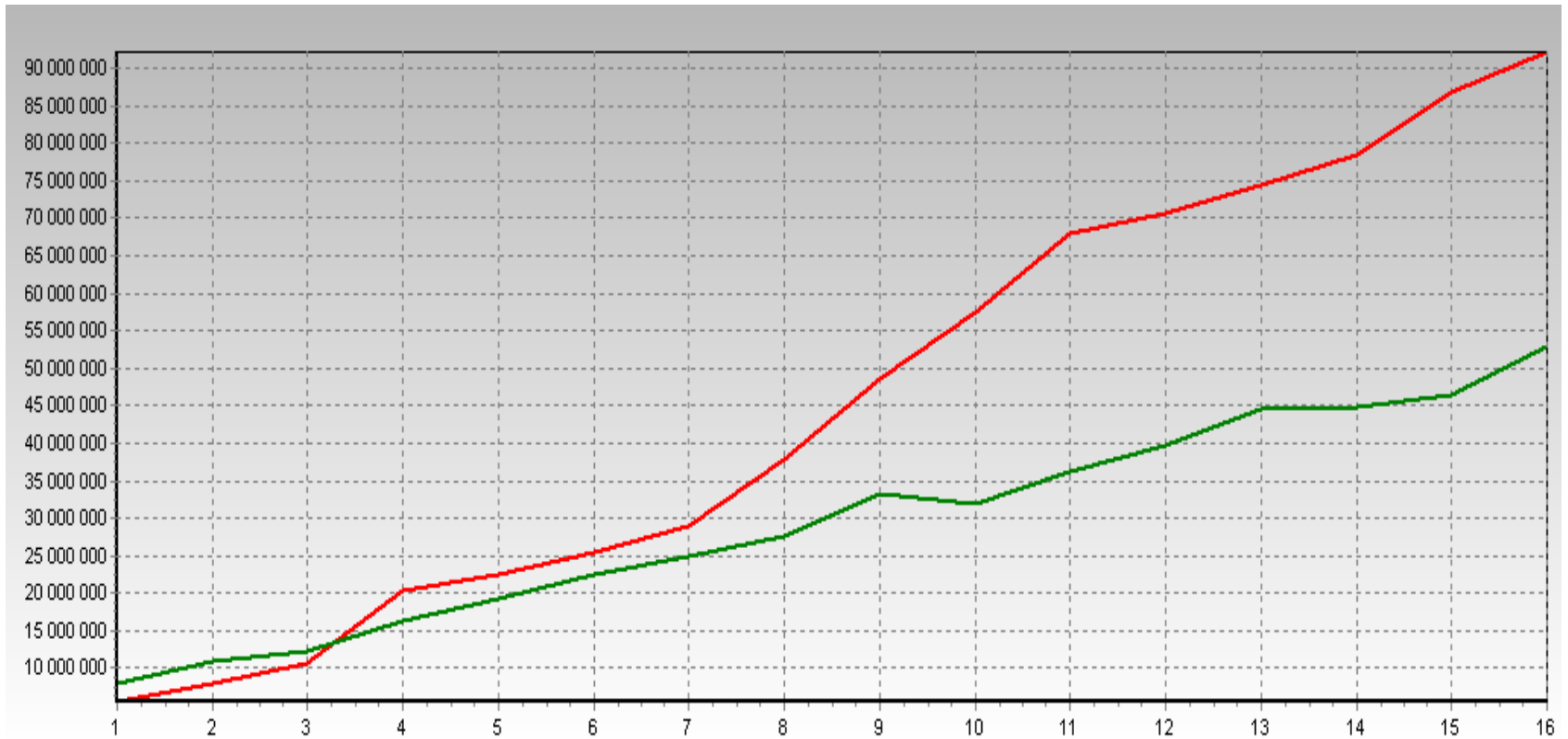
# Сравнение эффективности ссылочных и векторных структур



# Исследование эффективности предвыборки в TLB



# Использование оптимизирующих структур данных



# Конфликты в кэш-памяти

