



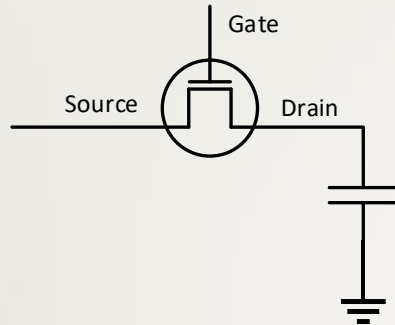
# Архитектура и организација рачунара 2

DRAM

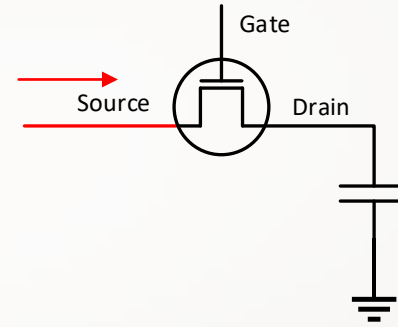


# DRAM

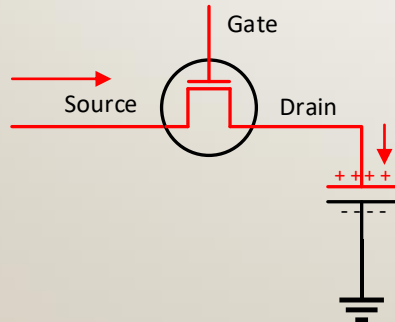
Основна меморијска јединица DRAM  
(транзистор + кондензатор = јефтино)



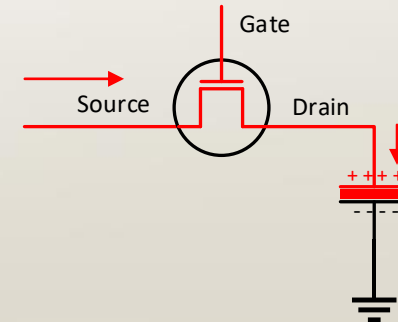
Gate је искључен, Source не пролази до Drain



Gate је укључен, Source пролази до Drain  
Кондензатор почиње да се пуни



Временом кондензатор се напуни

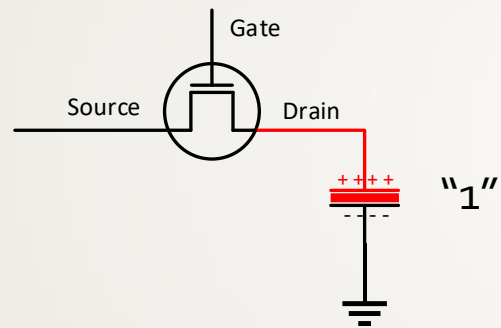


Напон логичке јединице  $\sim 3V$



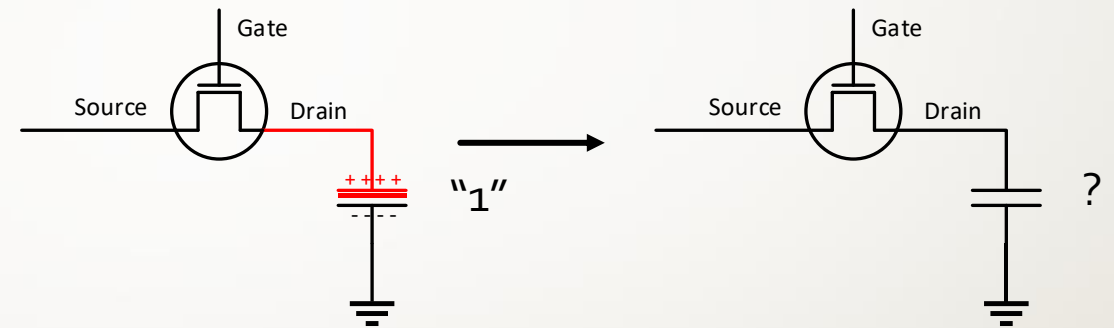
# DRAM

Ако се Gate искључи,  
стање кондензатора остаје исто  
(учитана логичка јединица)



На сличан начин се учитава и логичка нула

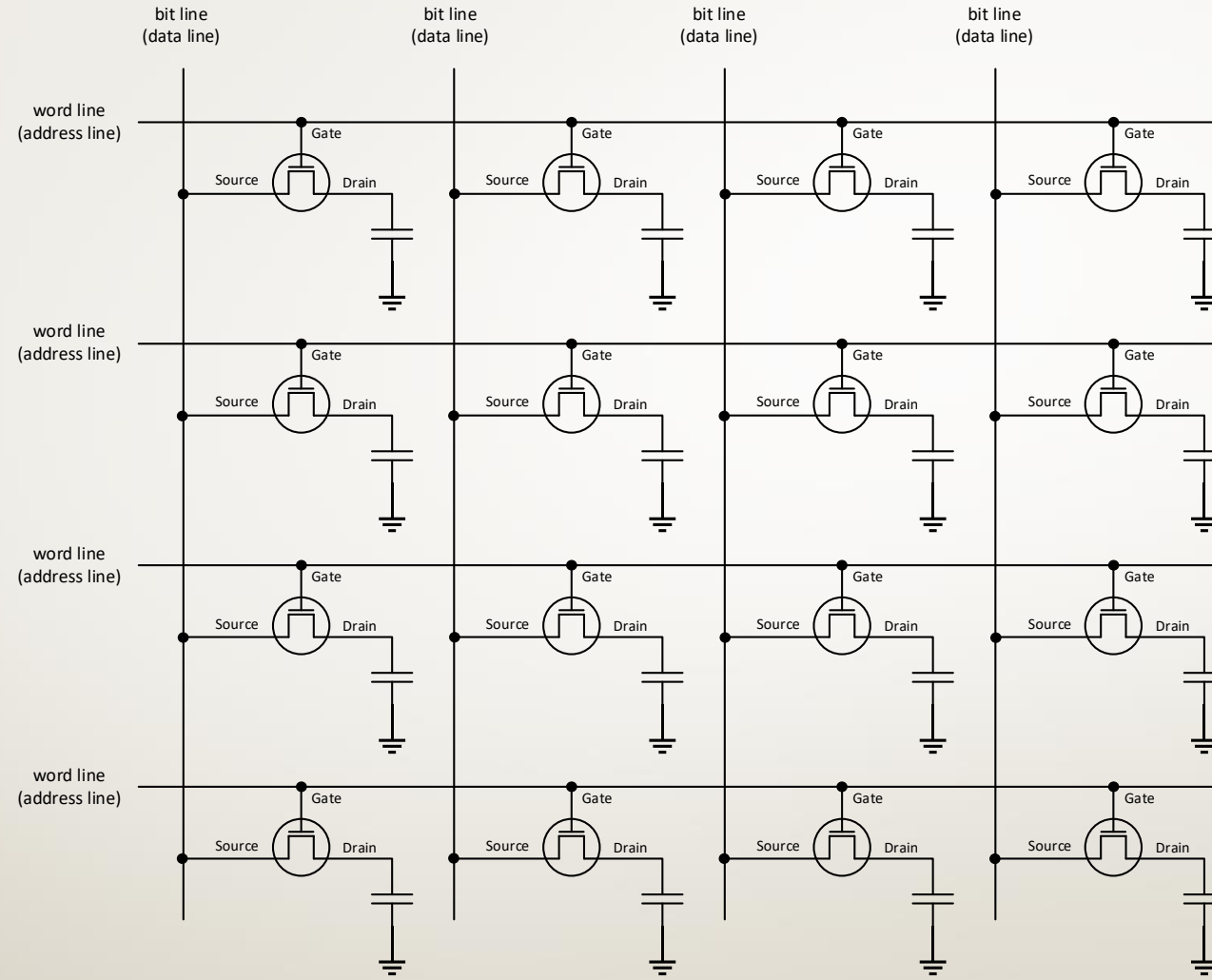
Међутим... Временом се кондензатор  
испразни...



**DRAM је потребно временом  
освежавати иначе изгуби вредност!**



# DRAM

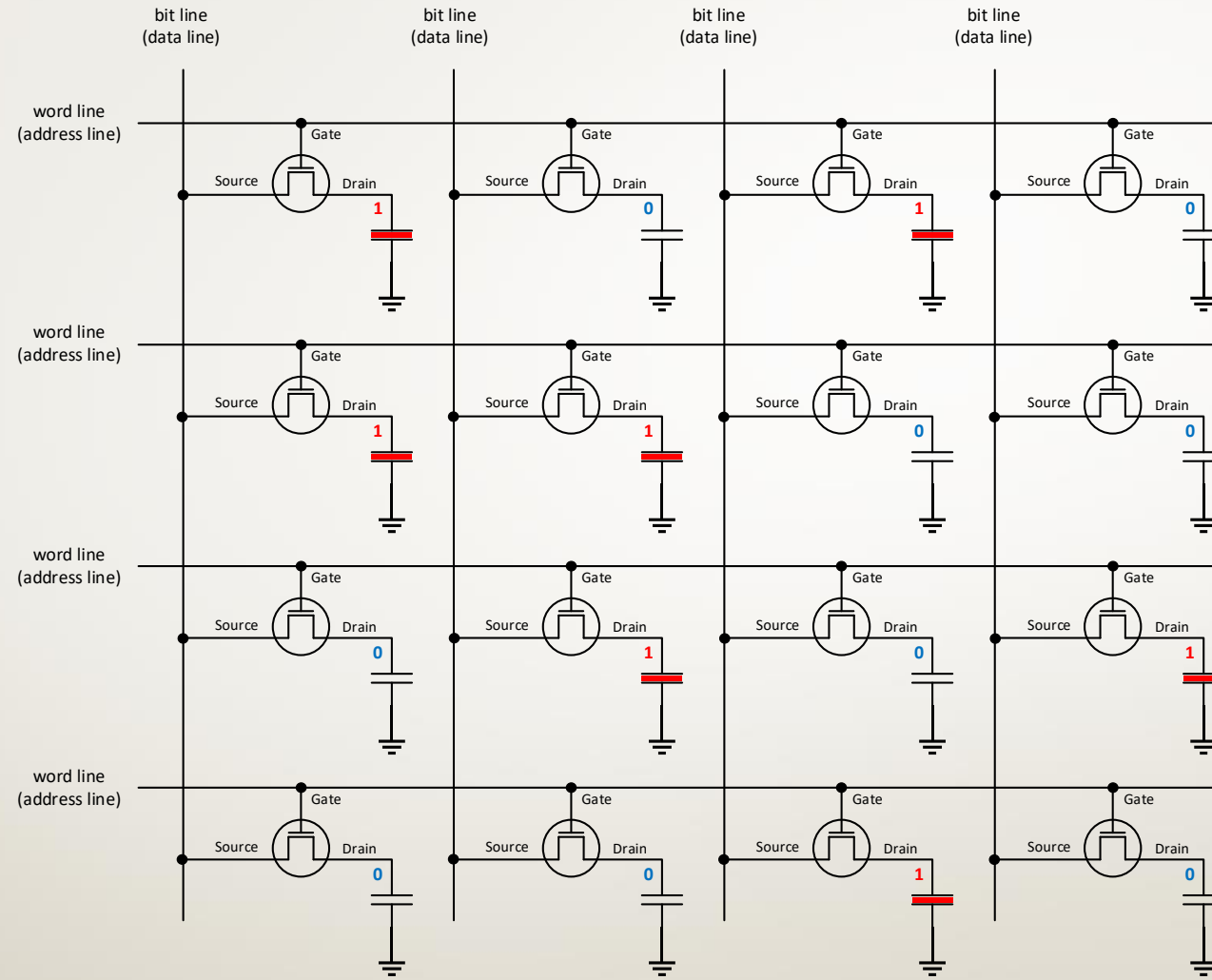


...

...



# DRAM

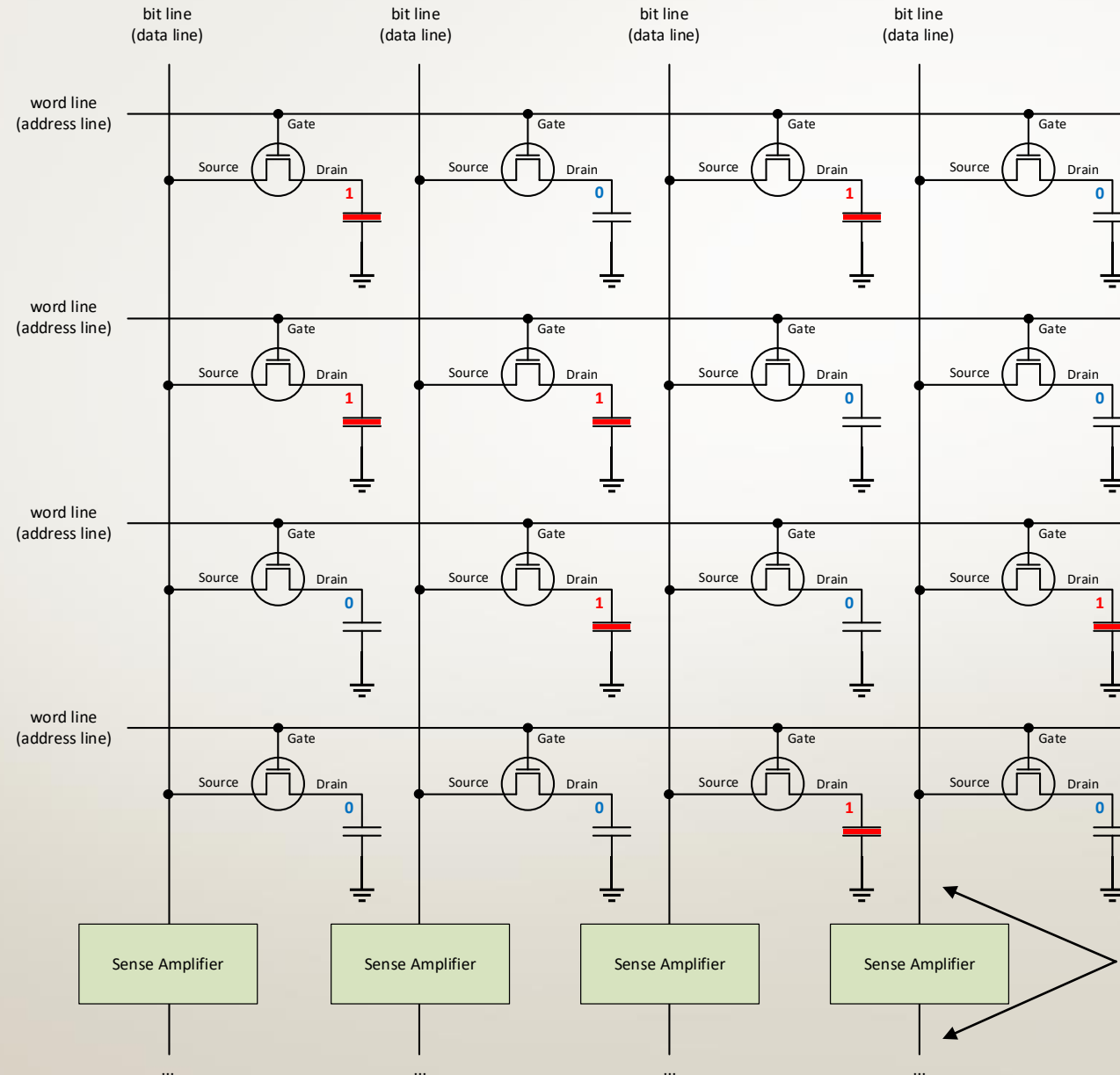


...

...



# DRAM



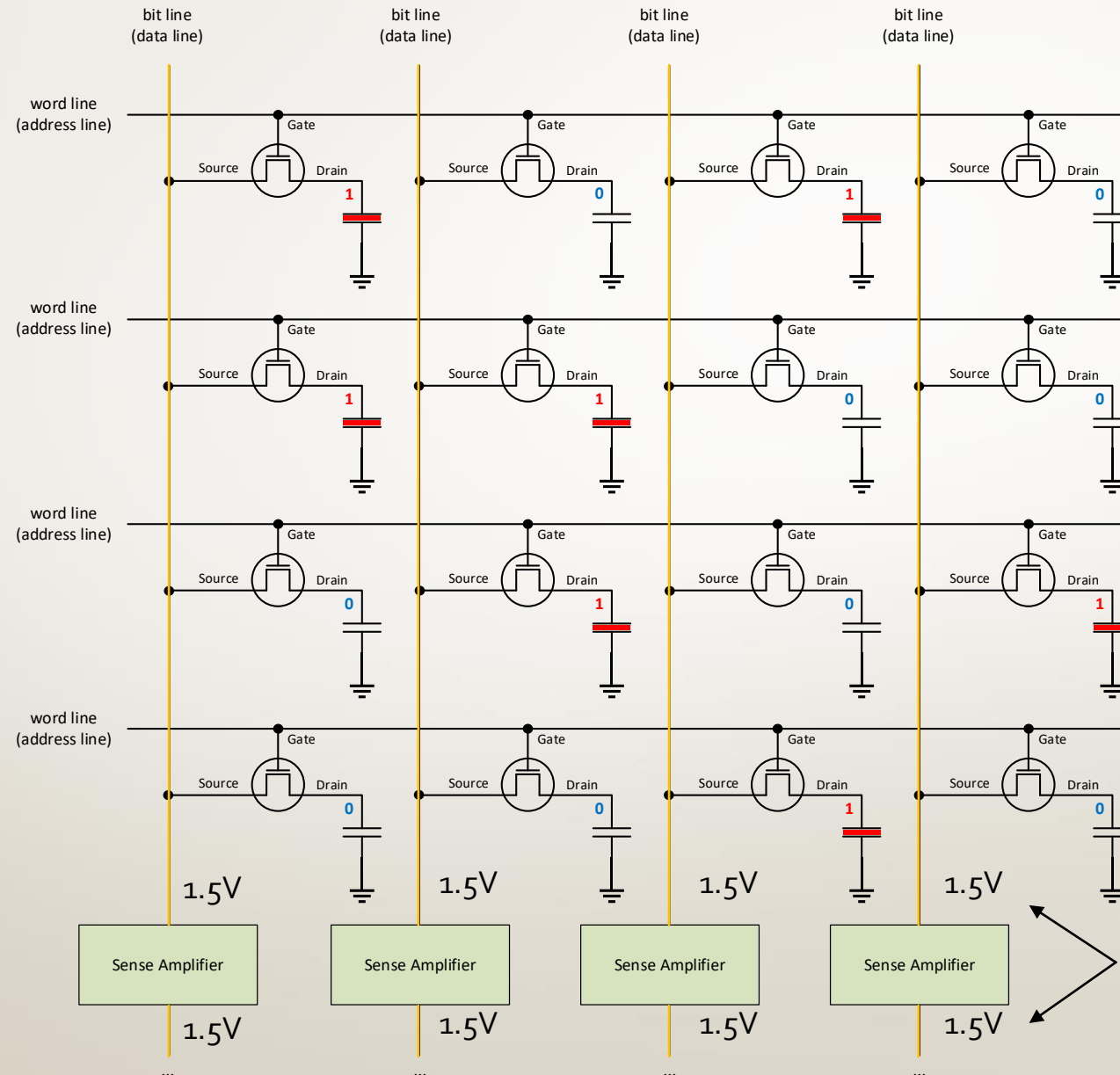
Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

Sense Amplifier упоређује  
напонске линије



# DRAM



Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

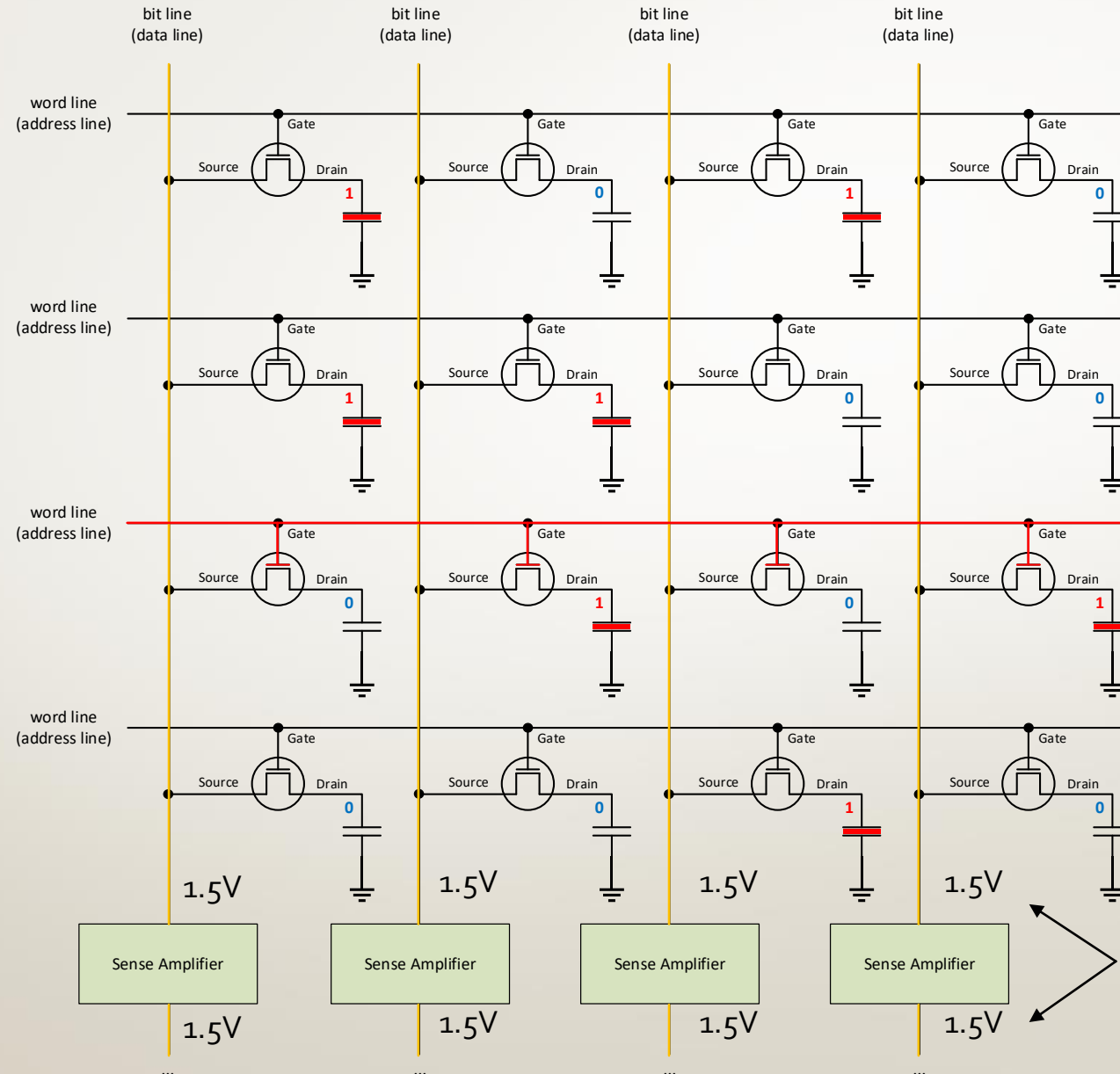
## PRECHARGE:

Поставља на bit line приближан напон половине збира напона логичке јединице и логичке нула (напон између њих)

Sense Amplifier упоређује напонске линије



# DRAM



Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

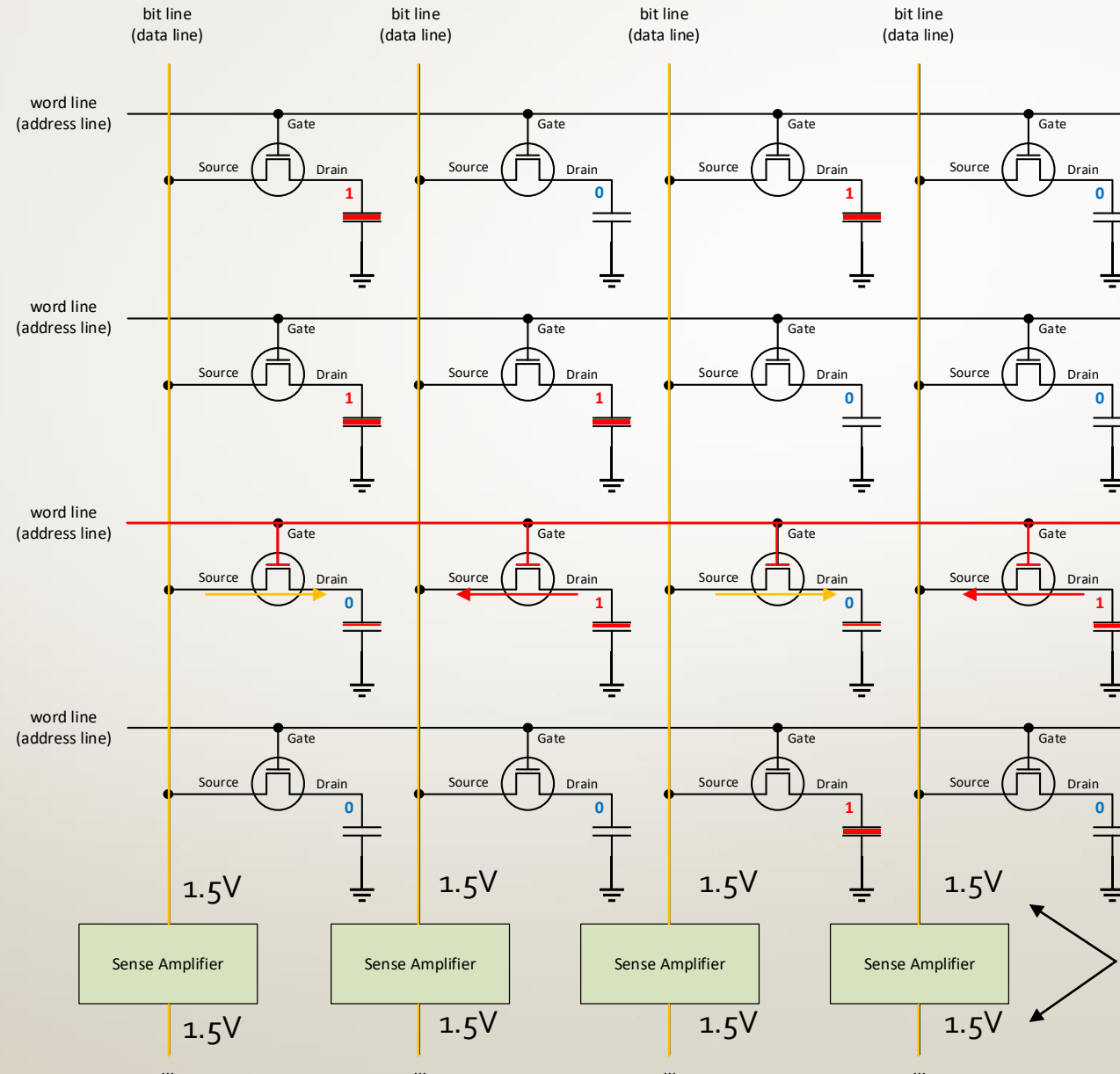
**RAS – Row Address Strobe**  
Активира се одговарајући ред  
(на основу дела адресе)

Sense Amplifier упоређује напонске линије





# DRAM



Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

**RAS – Row Address Strobe**

Активира се одговарајући ред  
(на основу дела адресе)

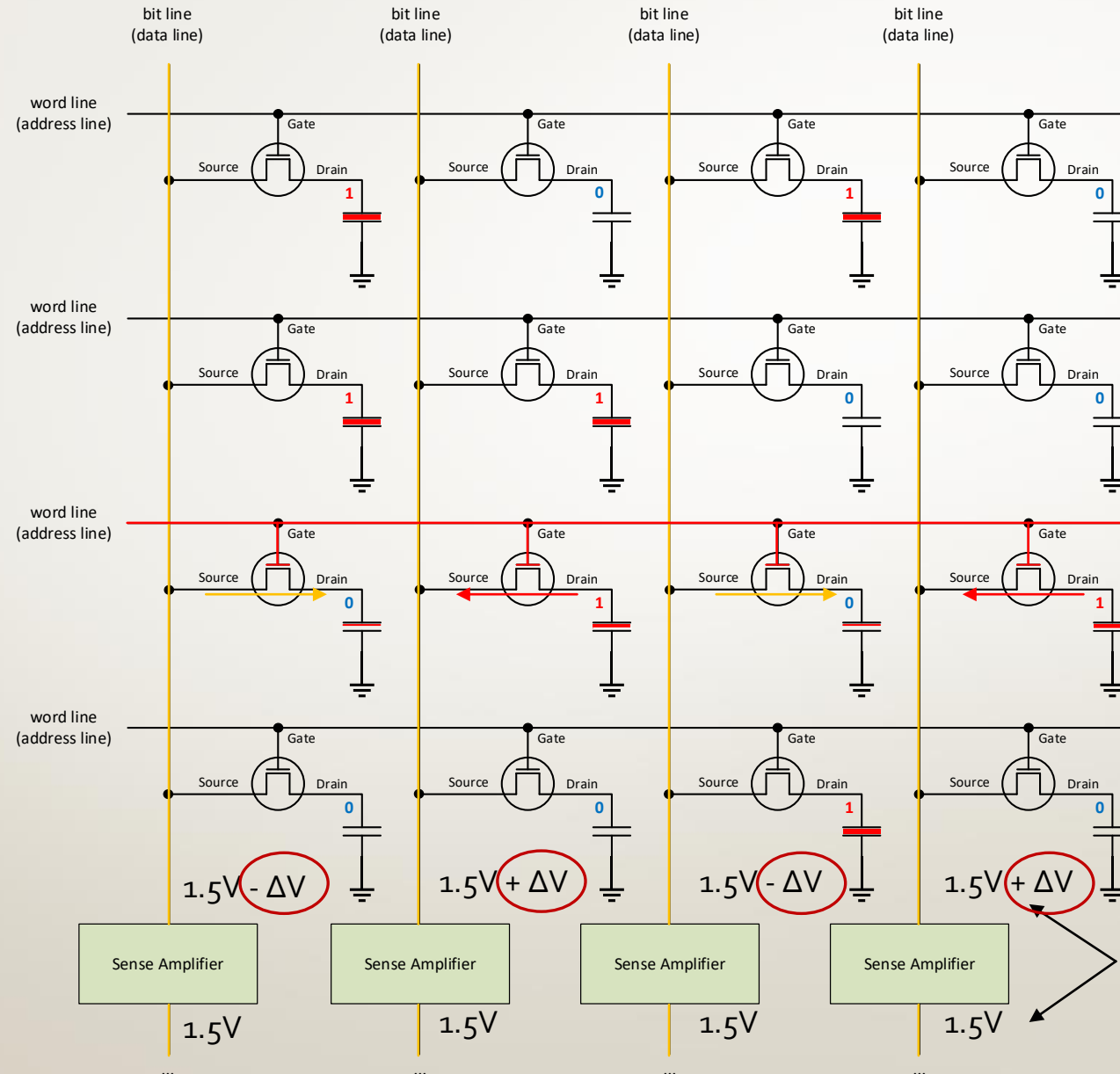
**Последица:**

Кондензатори логичке нуле примају наелектрисање;  
Кондензатори логичке јединице губе наелектрисање;

Sense Amplifier упоређује напонске линије



# DRAM



Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

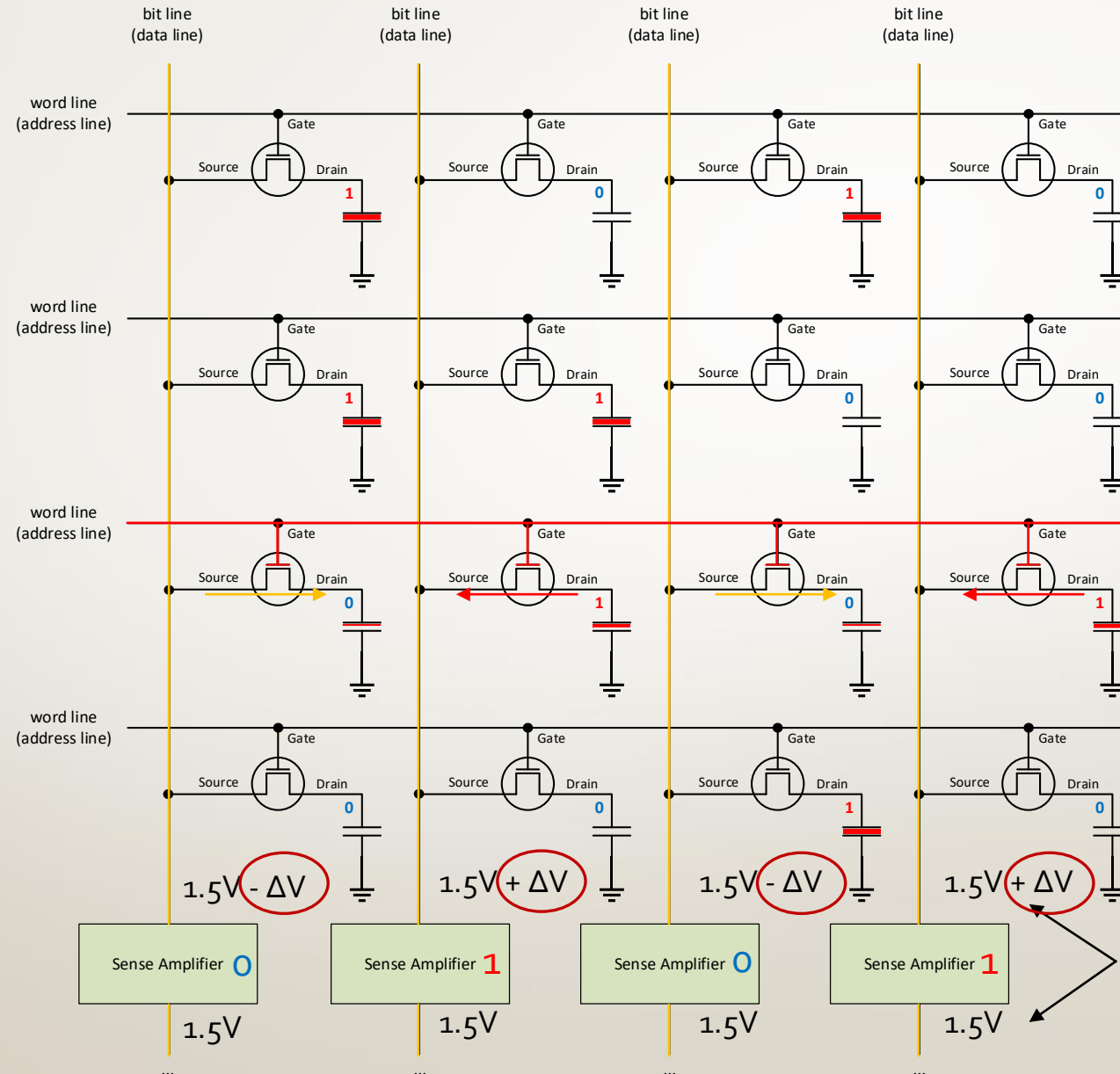
## RAS – Row Address Strobe

Ово стање јако кратко траје, али довољно да Sense Amplifier детектује промену у напону ( $\pm \Delta V$ ).

Sense Amplifier упоређује напонске линије



# DRAM



Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

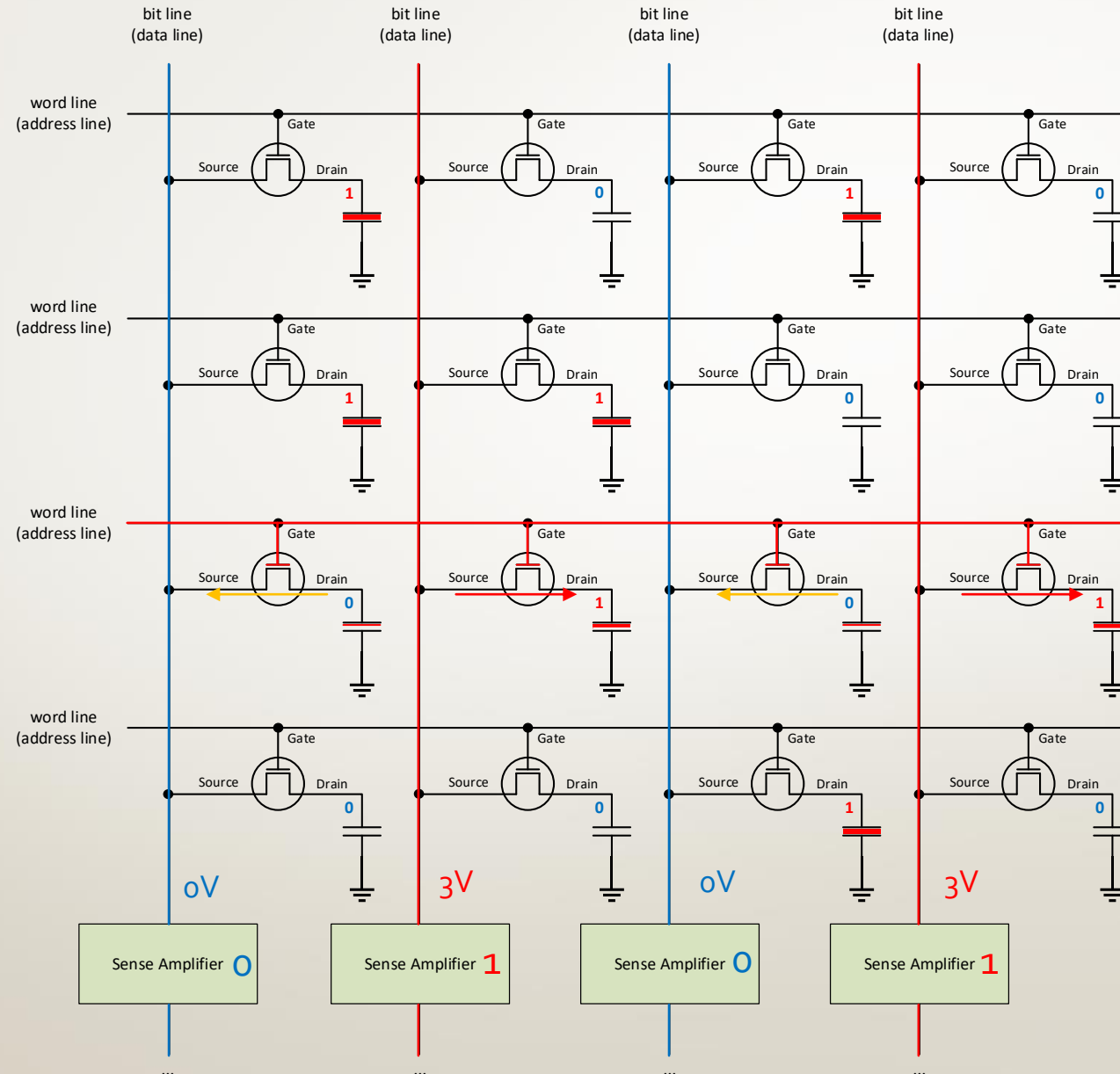
## RAS – Row Address Strobe

На основу промене напона Sense Amplifier учитава у неки флип-флоп логичку јединицу (код +  $\Delta V$ ), логичку јединицу (код -  $\Delta V$ )

Sense Amplifier упоређује напонске линије



# DRAM



Читање:

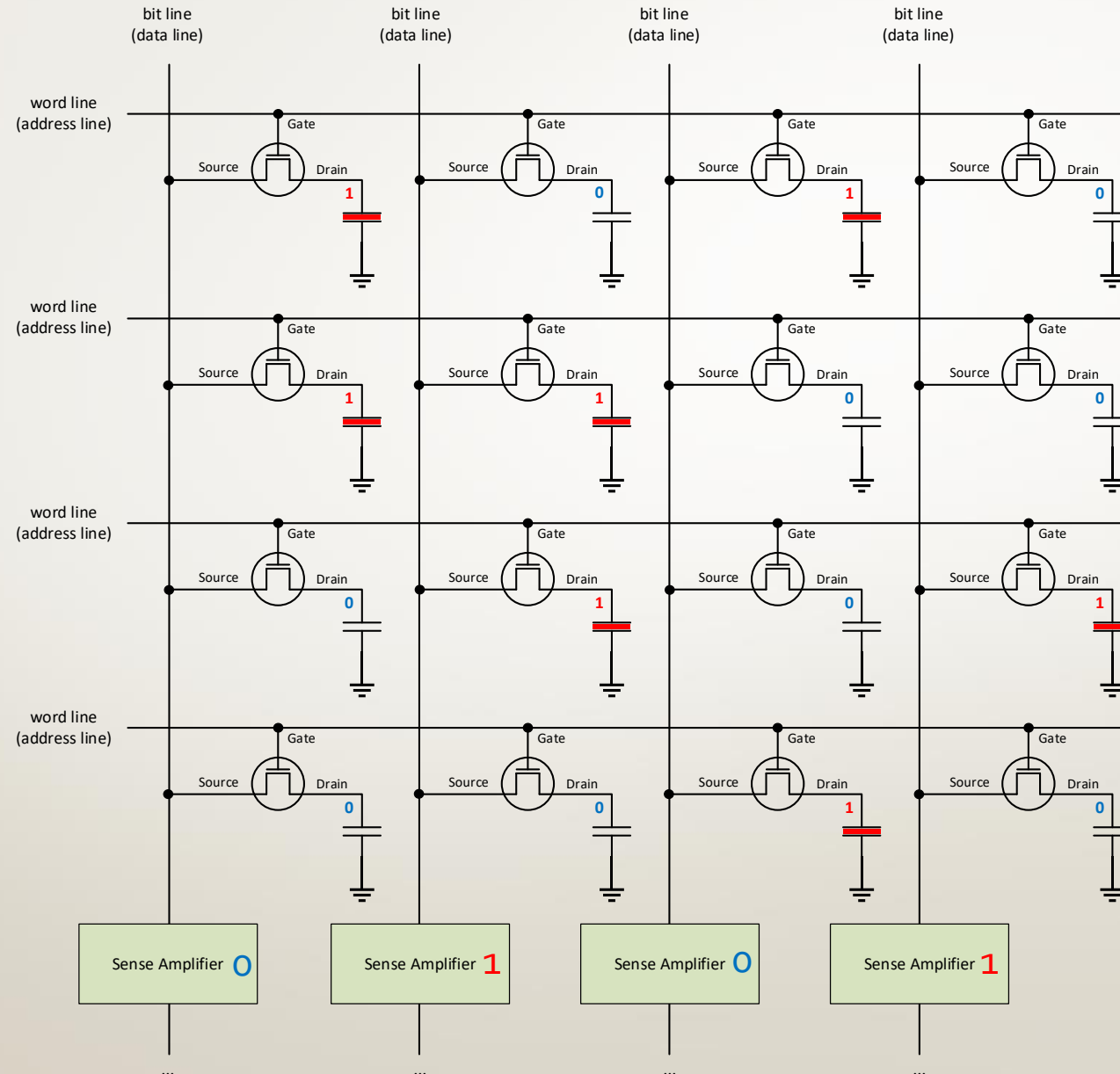
- 1) PRECHARGE
- 2) RAS
- 3) CAS

## RAS – Row Address Strobe

Потребно је вратити изгубљено/добијено наелектрисање у кондензаторима, тако да Sense Amplifier пушта прочитану вредност на bit line



# DRAM



Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

**RAS – Row Address Strobe**

Након враћеног наелектрисања повлачи се напон и са bit line и са word line

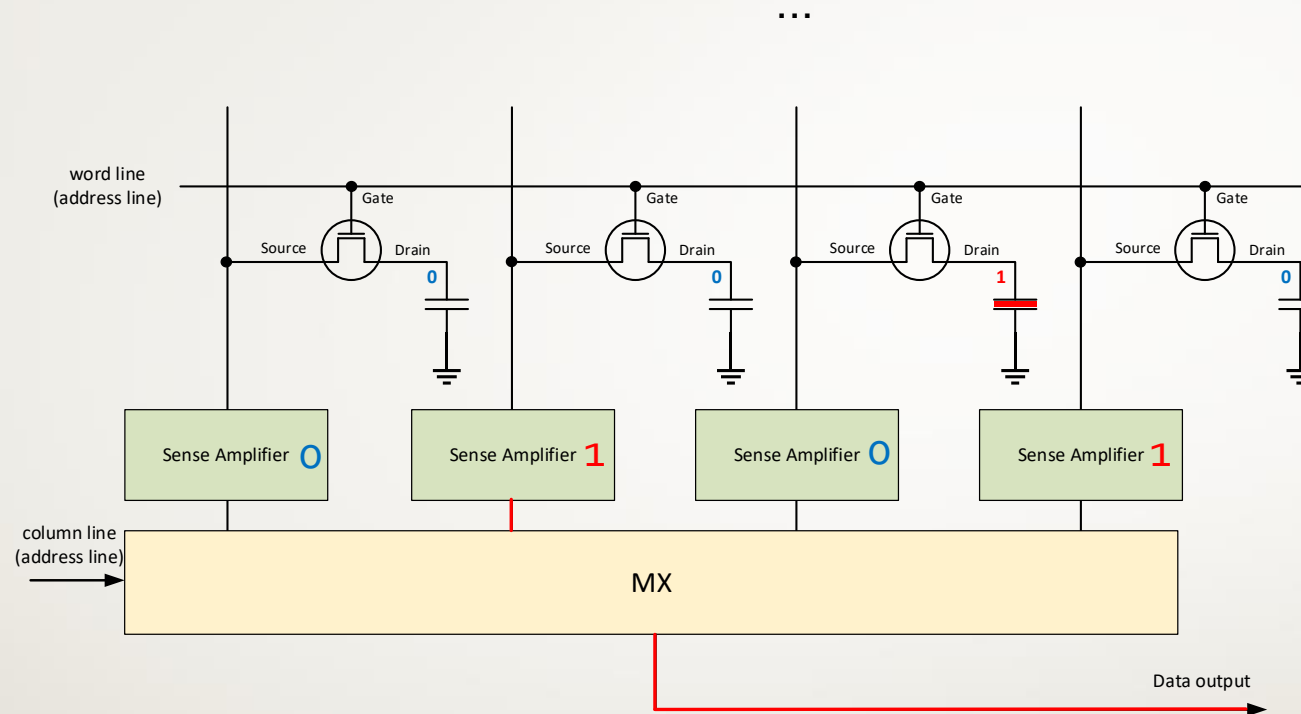
У Sense Amplifier остаје прочитана вредност



# DRAM

Читање:

- 1) PRECHARGE
- 2) RAS
- 3) CAS

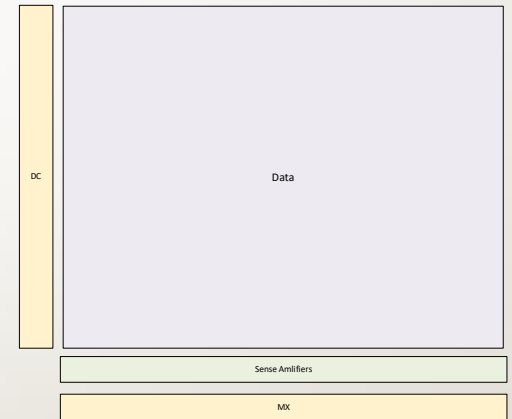
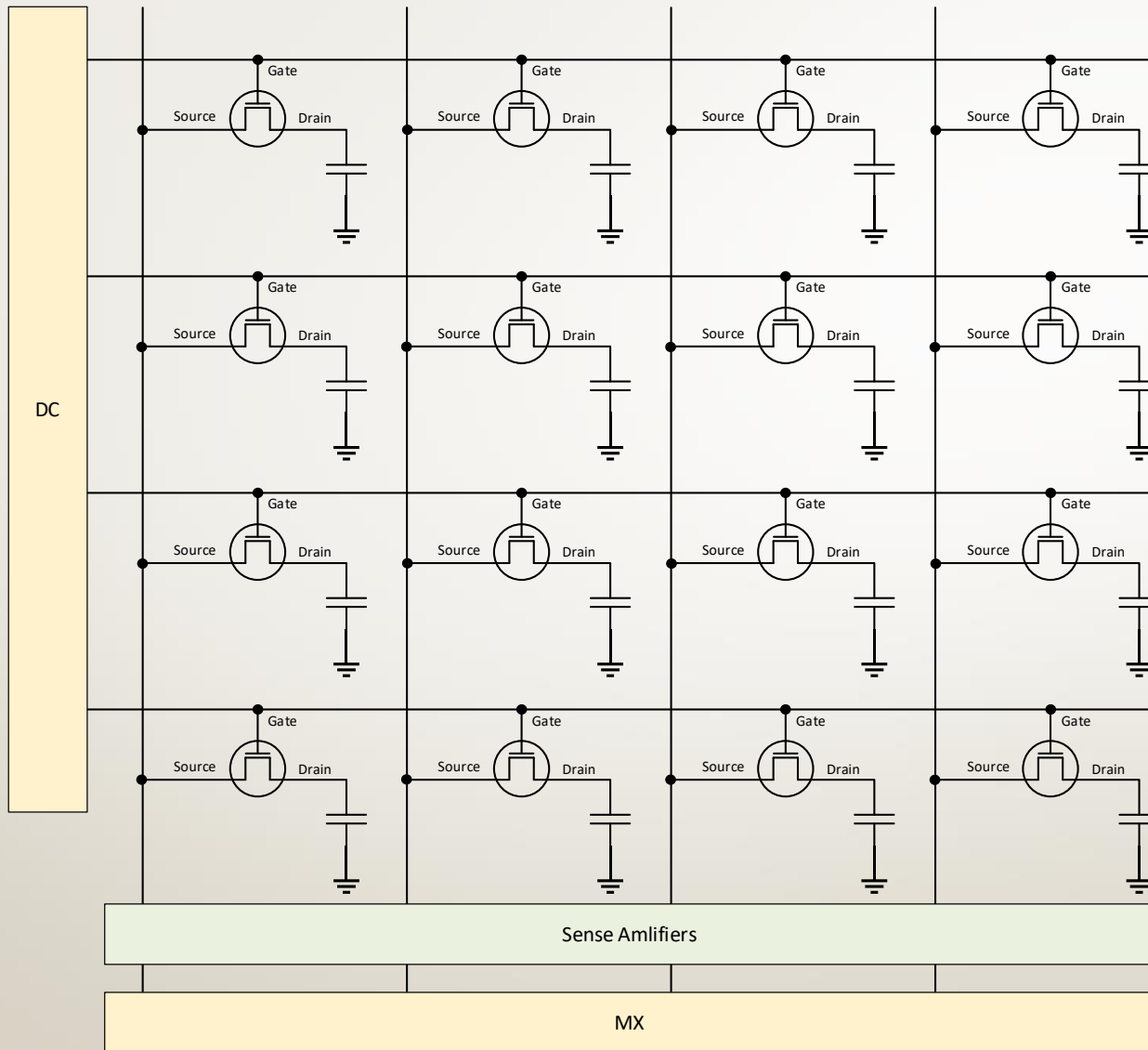


## CAS – Column Address Strobe

Преко мултиплексера, а на основу адресе, читају се подаци из Sense Amplifier.

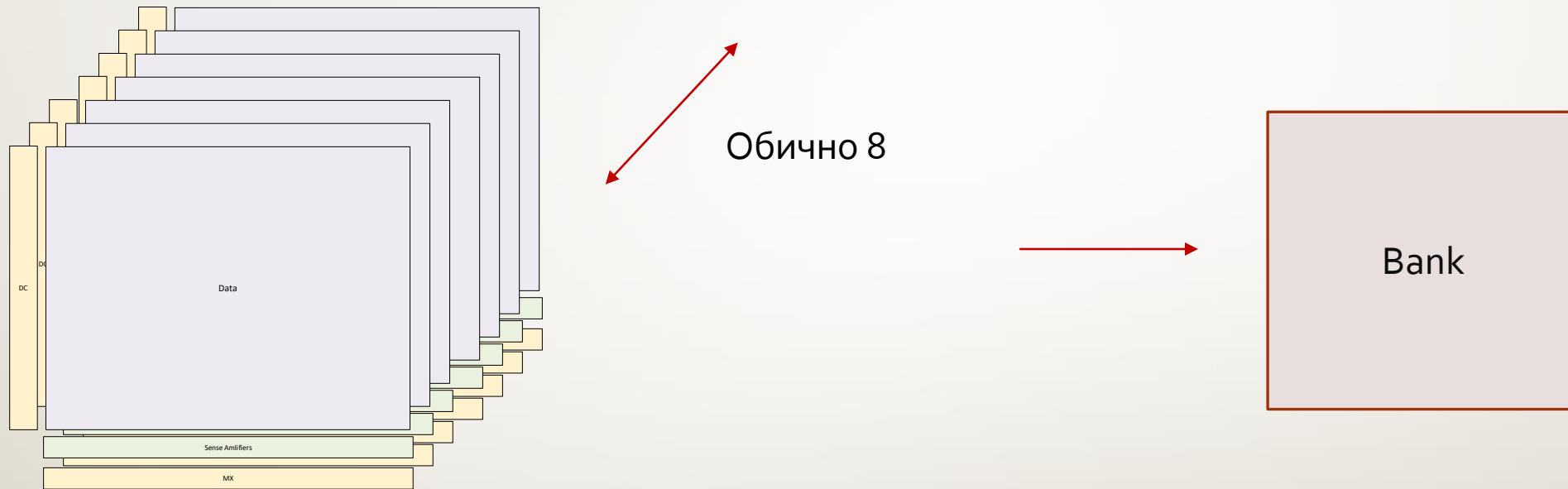


# DRAM





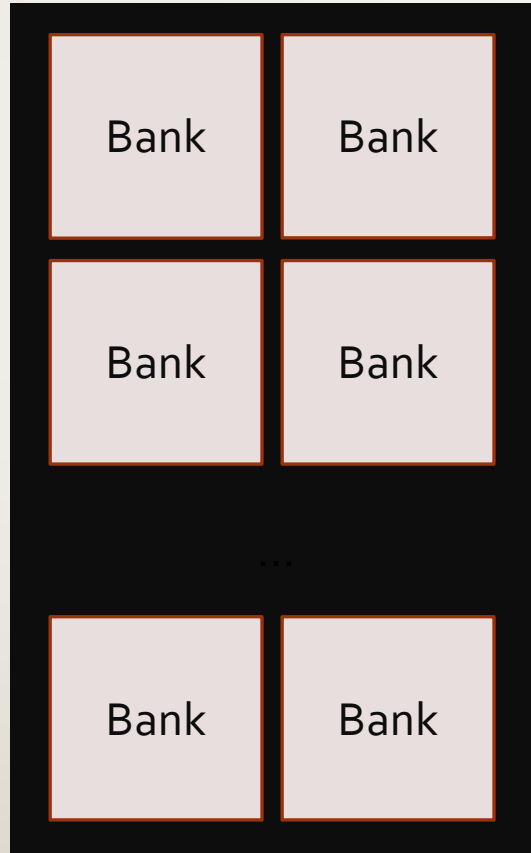
# DRAM







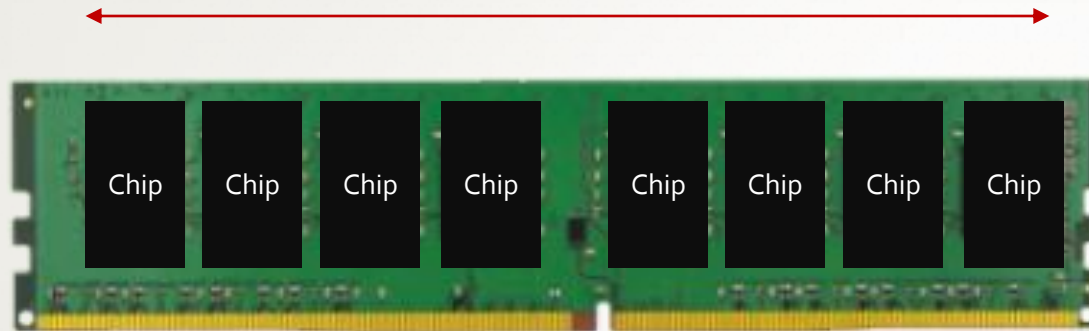
# DRAM





# DRAM

Обично:  $8 \times 8\text{bit} = 64 \text{ bit}$

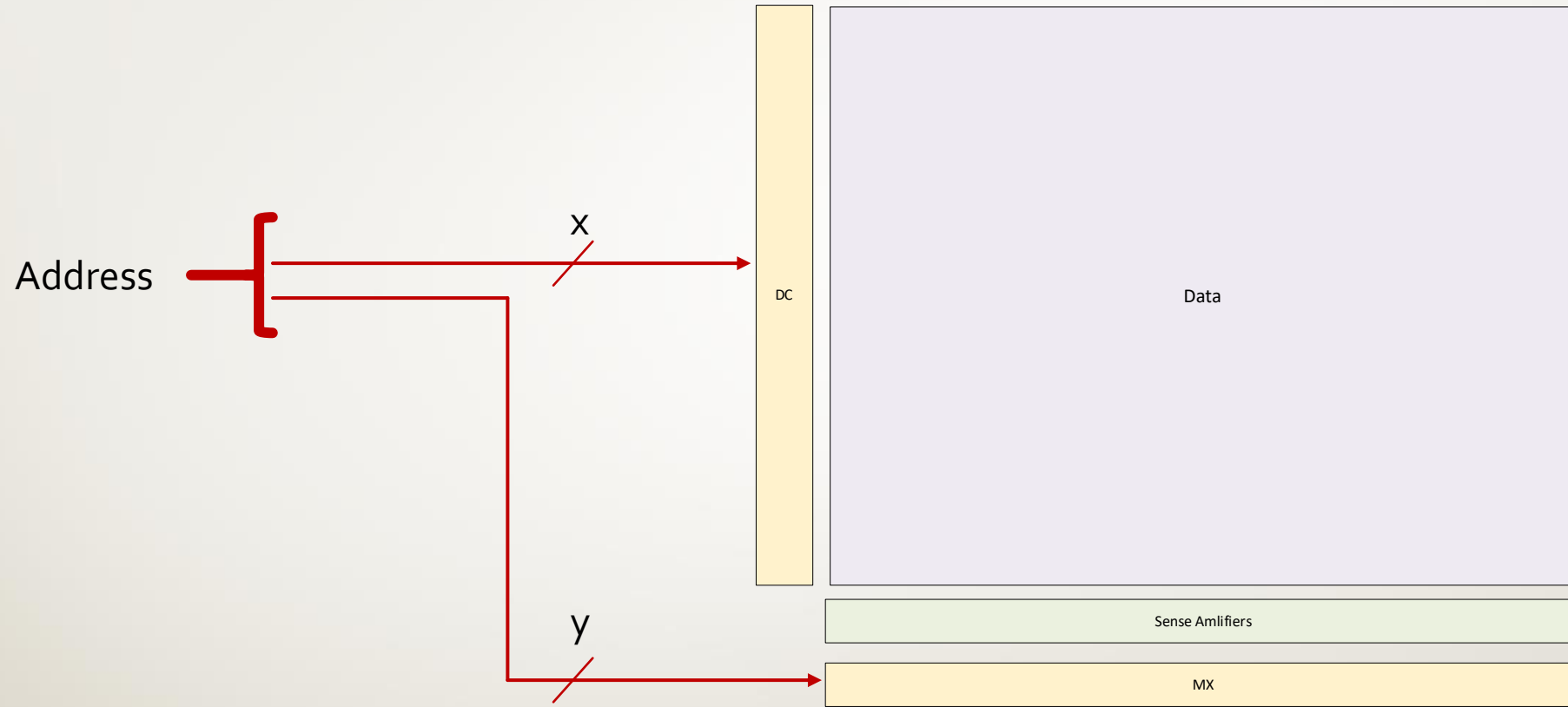


DIMM Module  
"RAM stick"

ECC RAM: RAM са Error Correction Code → *додатни chip-ови на DIMM*  
Обично сервери имају ECC



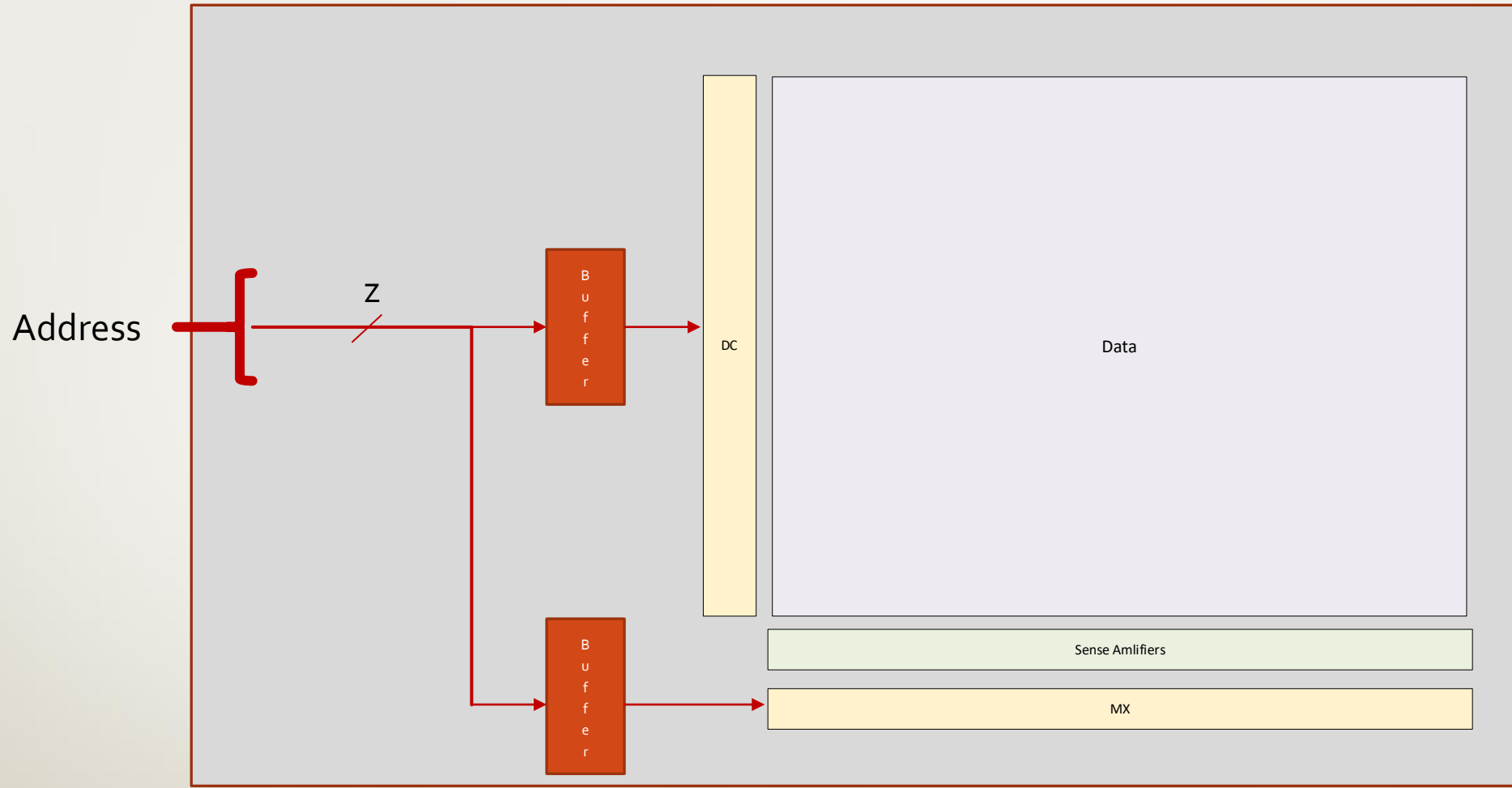
# DRAM



$x+y$  = ширина адресе



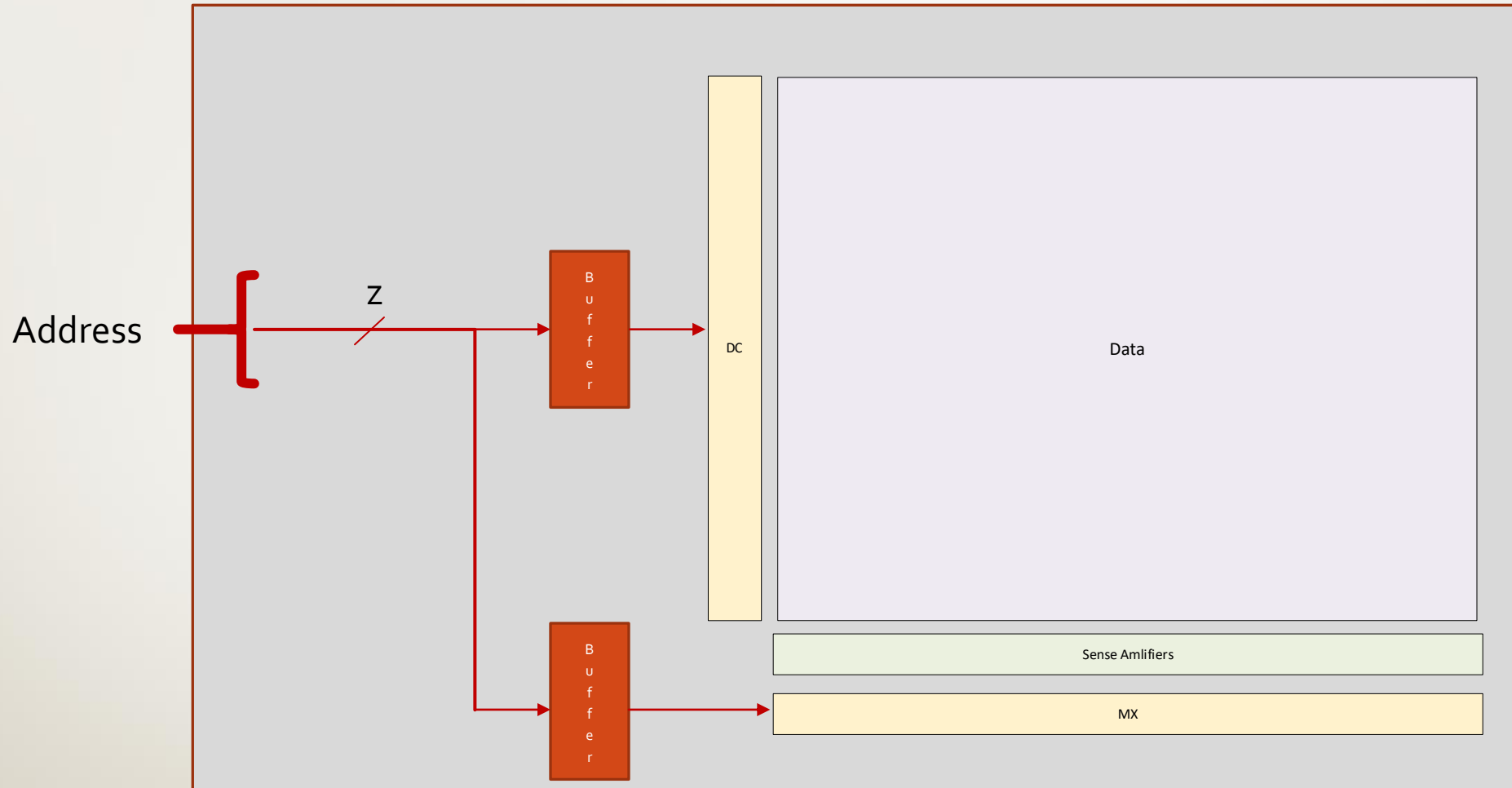
# DRAM



$$z < x + y$$



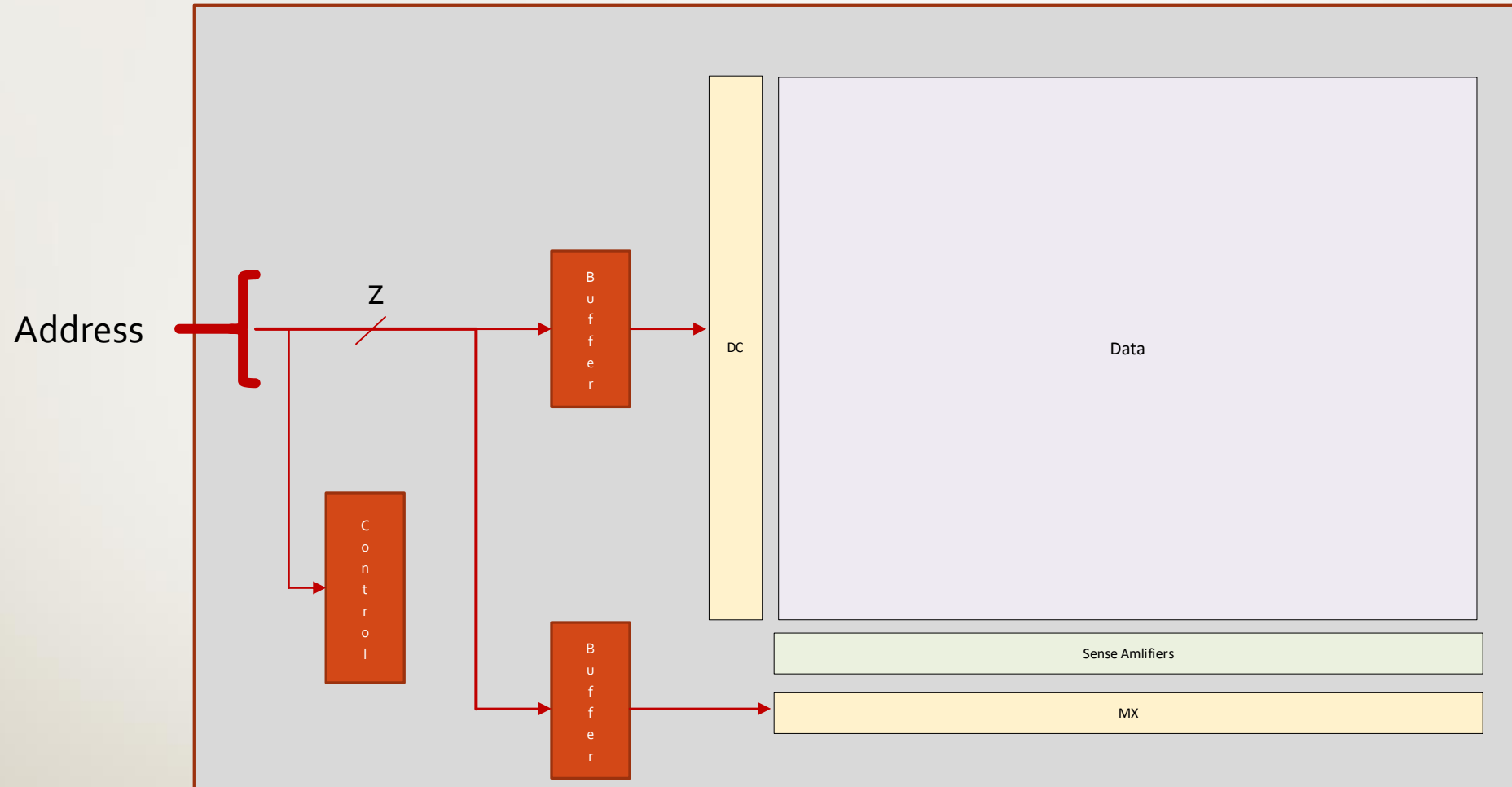
# DRAM



Како знам да ли читам-уписујем?  
Недостају контролни битови



# DRAM

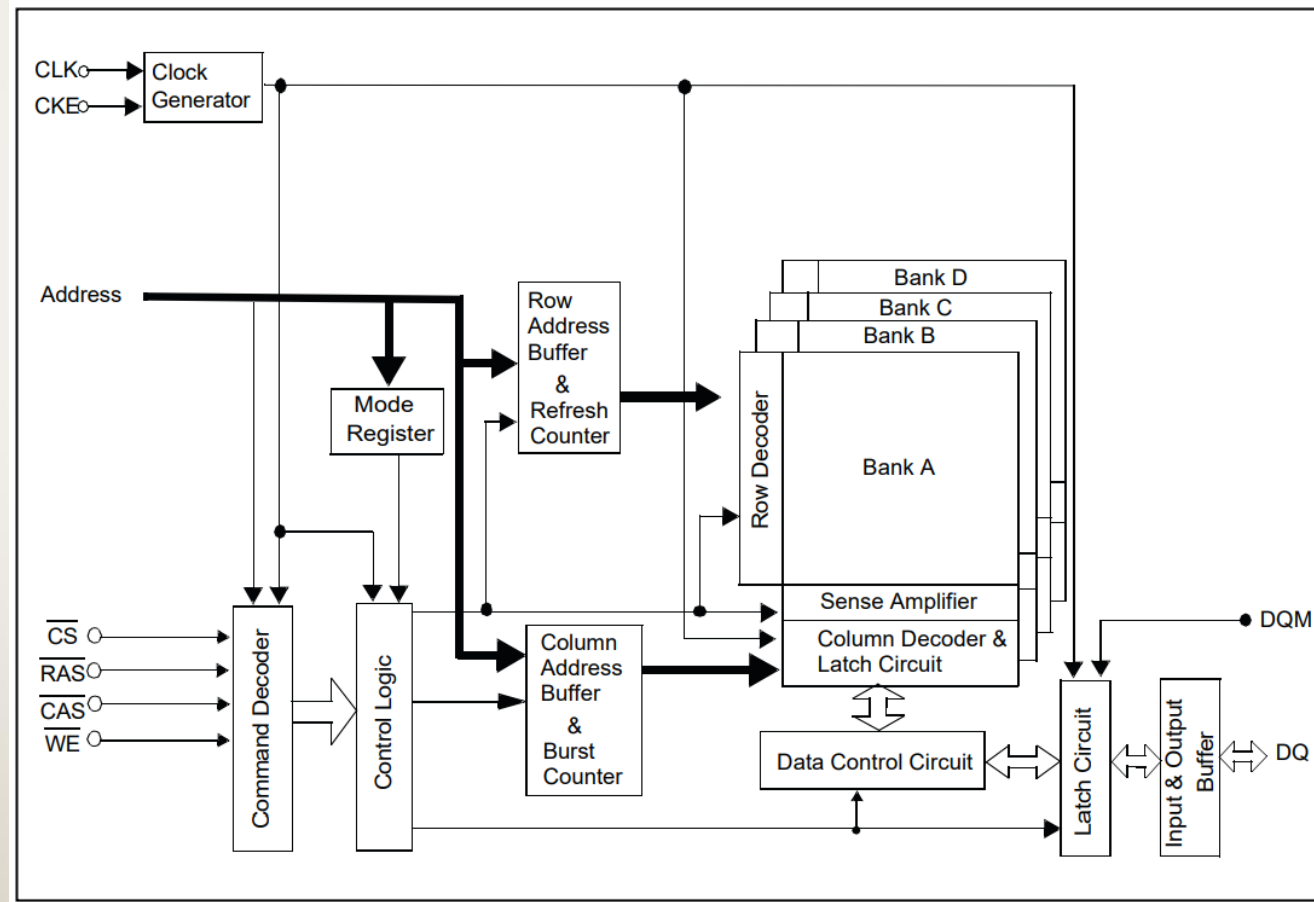


Уzeti битове адресе за контролне битове.  
Контролни битови: CS, RAS, CAS, WE



# DRAM

Блок дијаграм (документација чипа IS42S16400)





# Задатак 1

Дат је меморијски контролер који је повезан на DDR меморијски чип са 4 банке. Чип је ширине 16 бита. У наставку је описан рад контролера:

1. Меморијски контролер издаје наредбу за **Precharge** да би банка била спремна за приступ новом реду. **Precharge** је завршен након времена  $tRP$ .
2. Меморијски контролер затим издаје наредбу **Activate** за читање одговарајућег реда из банке. Команда Activate траје  $tRCD$  времена и након тога сматра се да је цео ред прочитан у бафер.
3. Меморијски контролер тада може издати наредбу за приступ колони (CAS) и том приликом чита податке из бафера и ставља их на меморијски канал. После времена  $tCL$ , првих 64 бита података су постављени на меморијски канал. Контролер чита кружно осам сукцесивних локација (што представља и величину бафера) од адресе која је задата, а затим аутоматски враћа бафер у банку (**Auto precharge**).
4. Ако контролер жели да приступи подацима у другом реду банке, он понавља кораке (1) - (3). Ако меморијски контролер жели да приступи другом блоку података у истом реду, који се налази унутар бафера, једноставно издаје другу CAS команду.
5. Меморијски контролер може издавати команде различитим банкама у низу тако да могу да изводе читања / писања паралелно.
6. Потребно је обезбедити да је минимални временски размак између Activate команде и наредне Precharge команде једнак  $tRAS$ .





# Задатак 1

Претпоставимо да је  $t_{PC}=t_{RCD}=t_{CL}=13$  ns и  $t_{RAS}=40$  ns, и да је фреквенција меморијског канала 1GHz (Data rate clock).

- I. После ког времена најраније се може прочитати податак DATA из меморије, ако се тренутни ред који одговара податку DATA налази у баферу?
- II. После ког времена најраније може се прочитати податак DATA из меморије, ако се тренутни ред који одговара податку DATA **не** налази у баферу?
- III. После ког времена најраније може се прочитати читав ред из меморије, ако се тренутни ред не налази у баферу?
- IV. Како би требало распоредити податке по банкама да би се најбрже добили сукцесивних 20 адреса. После ког се могу прочитати тих 20 адреса ако су сви бафери банака празни?



# Задатак 1

Претпоставимо да је  $t_{PC}=t_{RCD}=t_{CL}=13\text{ ns}$  и  $t_{RAS}=40\text{ ns}$ , и да је фреквенција меморијског канала  $1\text{GHz}$  (Data rate clock).

- I. После ког времена најраније се може прочитати податак DATA из меморије, ако се тренутни ред који одговара податку DATA налази у баферу?

*Податак је прочитан у бафер → преостаје да се само још прочита из бафера, тј. треба само да се изда команда за читање по колони CAS. Време за читање по колони је  $t_{CL} = 13\text{ns}$*

- II. После ког времена најраније може се прочитати податак DATA из меморије, ако се тренутни ред који одговара податку DATA **не** налази у баферу?

*За читање податка потребно је одрадити:*

- 1) Precharge –  $t_{PC}$*
- 2) Активирати ред –  $t_{RCD}$*
- 3) Прочитати из колоне –  $t_{CL}$*

**Одговор:  $t_{PC} + t_{RCD} + t_{CL}$**

**Дискусија:** није назначено у тексту да ли је меморија већ у стању Precharge, ако јесте можемо да прескочимо први корак, па уопштено записујемо:  **$(t_{PC}) + t_{RCD} + t_{CL}$**



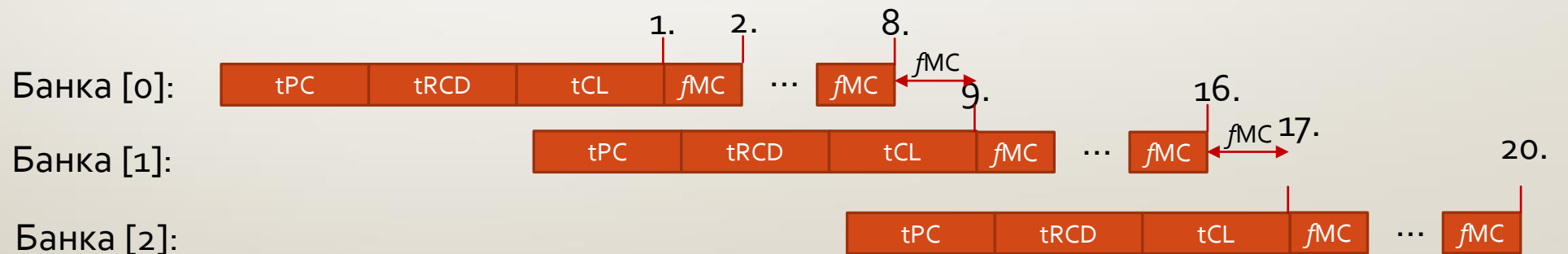
# Задатак 1

Претпоставимо да је  $t_{PC}=t_{RCD}=t_{CL}=13$  ns и  $t_{RAS}=40$  ns, и да је фреквенција меморијског канала 1GHz.

III. После ког времена најраније може се прочитати читав ред из меморије, ако се тренутни ред не налази у баферу?

Величина бафера = 8 и на основу решења II  $\rightarrow (t_{PC}) + t_{RCD} + t_{CL} + 7 * (1/1GHz)$

IV. Како би требало распоредити податке по банкама да би се најбрже добили сукцесивних 20 адреса. После ког се могу прочитати тих 20 адреса ако су сви бафери банака празни?



Одговор:  $t_{PC}+t_{RCD}+t_{CL}+19*(1/1GHz)$



# Задатак 1

Претпоставимо да је  $t_{PC}=t_{RCD}=t_{CL}=13$  ns и  $t_{RAS}=40$  ns, и да је фреквенција меморијског канала 1GHz.

- III. Како би требало распоредити податке по банкама да би се најбрже добили сукцесивних 20 адреса. После ког се могу прочитати тих 20 адреса ако су сви бафери банака празни?

Да ли након коришћења банке [1] можемо опет да наставимо са банком [0]?

Помоћ:



Мора важити:  $t > t_{RAS}$

\* Напомена за обележавања у различитим литературама и документацијама меморијских чипова:

$t_{PC} = t_{RP}$

$t_{CL} = CAS$



- Потребно је израчунати следећа времена DDR<sub>4</sub> RAM за задате меморије.
  - Времена:
    - После ког времена најраније може се прочитати податак DATA из меморије, ако се тренутни ред који одговара податку DATA **не** налази у баферу?  
 **$tPC + tRCD + tCL$**
    - После ког времена најраније може се прочитати читав ред из меморије, ако се тренутни ред не налази у баферу? DDR<sub>4</sub> дефинише A<sub>0</sub>-A<sub>9</sub> за колоне, што значи да један ред код DDR<sub>4</sub> има 2<sup>10</sup> података = 1024 (ширина Sense Amplifier)  
 **$(tPC) + tRCD + tCL + 1024 * (\text{такт преноса})$**
  - RAM:
    - CMW16GX<sub>4</sub>M1D3000C16 ([link](#))
    - Kingston HX440C19PB3/8 ([link](#))
    - Patriot PVSR48G360Co ([link](#))
    - RAM меморија која се већ налази на рачунару (користити софтвер CPU-Z)



## Задатак 2

- CMW16GX4M1D3000C16 ([link](#))
  - Ознака кашњења:
    - Base: 15-15-15-36
    - Tested: 16-20-20-38
- Kingston HX440C19PB3/8 ([link](#))
  - Ознака кашњења:
    - JEDEC: 17-17-17
    - XMP #1: 19-21-21
    - XMP #2: 17-18-18
- Patriot PVSR48G360Co ([link](#))
  - Ознака кашњења:
    - Base: 19-19-19-43
    - Tested: 20-26-26-46
- RAM меморија која се већ налази на рачунару (користити софтвер CPU-Z)

Ознака кашњења:  
CAS – tRCD – tRP – tRAS



# Задатак 2

Р.Б.	Модел	Профил	Ознака кашњења	Data rate clock	I/O bus clock	Цена	CAS		tRCD		tRP		tRAS		Први податак	Ред	Кеш линија (64В)
							cycle	ns	cycle	ns	cycle	ns	cycle	ns			
1	CMW16GX4M1D3000C16	Base	15-15-15-36	2133MHz	1066.5MHz	\$84.99	15	14.0646976	15	14.06469761	15	14.06469761	36	33.75527426	42.19409283	521.8002813	45.4758556
		Tested	16-20-20-38	3000MHz	1500MHz		16	10.6666667	20	13.33333333	20	13.33333333	38	25.33333333	37.33333333	378.3333333	39.66666667
2	HX440C19PB3/8	JDEC	17-17-17	2400MHz	1200MHz	\$94	17	14.1666667	17	14.16666667	17	14.16666667	32		42.5	468.75	45.41666667
		XMP #1	19-21-21	4000Mhz	2000MHz		19	9.5	21	10.5	21	10.5			30.5	286.25	32.25
		XMP #2	17-18-18	3600MHz	1800MHz		17	9.44444444	18	10	18	10			29.44444444	313.6111111	31.38888889
3	PVSR48G360C0	Base	19-19-19-43	2666MHz	1333MHz	\$63	19	14.2535634	19	14.25356339	19	14.25356339	43	32.25806452	42.76069017	426.4816204	45.38634659
		Tested	20-26-26-46	3600MHz	1800MHz		20	11.11111111	26	14.44444444	26	14.44444444	46	25.55555556	40	324.1666667	41.94444444