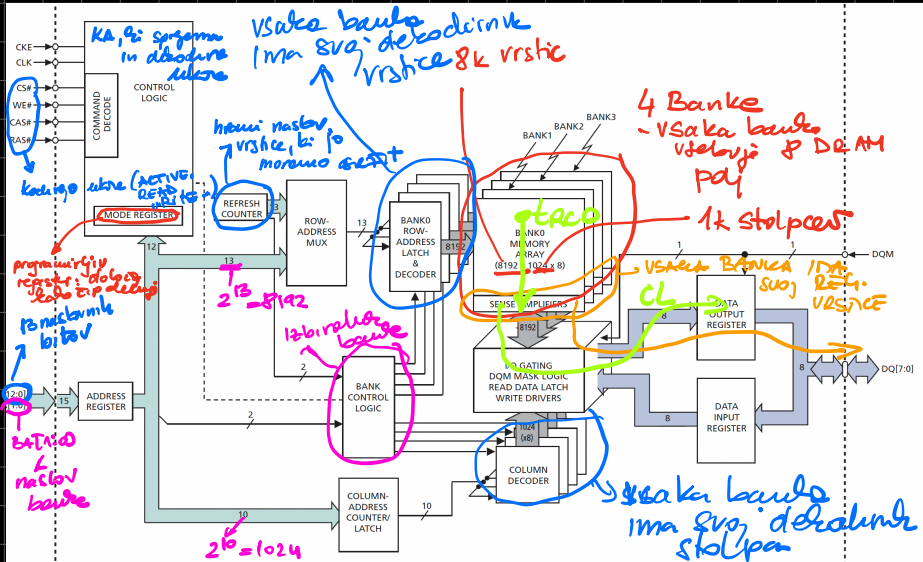



1 DRAM celica hrani informaciju najviše 64 ms

1x na 64 ms jo manas
osvieda

Nacelot laho osivmo 1024 stolpcer = celo vrstko

✓ vsrpa skupaj je 4×8192 vrstic = 32K

$$2^{15} = 32768$$

$$\rightarrow \text{periode osretuvanya} = \frac{64 \text{ ms}}{2^{15}}$$

Casi :

Speed Grade	Clock Frequency (MHz)	Target t_{RCD}^{tRP-CL}	t_{RCD} (ns)	t_{RP} (ns)	CL (ns)
-6A	167	3-3-3	18	18	18
-75	133	3-3-3	20	20	20
-7E	133	2-2-2	15	15	15

7.5ns

čas v urinskih periodah

ACTIVE-to-PRECHARGE command	t_{RAS}	42	120,000	37	120,000	44	120,000	ns	
ACTIVE-to-ACTIVE command period	t_{RC}	60	–	60	–	66	–	ns	11
ACTIVE-to-READ or WRITE delay	t_{RCD}	18	–	15	–	20	–	ns	
Refresh period (8192 rows)	t_{REF}	–	64	–	64	–	64	ms	
Refresh period – automotive (8192 rows)	t_{REFAT}	–	16	–	16	–	16	ms	
AUTO REFRESH period	t_{RFC}	60	–	66	–	66	–	ns	
PRECHARGE command period	t_{RP}	18	–	15	–	20	–	ns	
ACTIVE bank a to ACTIVE bank b command	t_{RRD}	12	–	14	–	15	–	ns	
Transition time	t_T	0.3	1.2	0.3	1.2	0.3	1.2	ns	12
WRITE recovery time	t_{WR}	1 CLK + 6ns	–	1 CLK + 7ns	–	1 CLK + 7.5ns	–	ns	13
		12	–	14	–	15	–	ns	14

t_{RCD} : čas od izstavitve ACTIVE ukaza do izstavitve READ/WRITE ukaza

→ čas potreben, da se informacija iz vrstice v banki zapiše v rep. vrstice (= tipalni gajracikli)

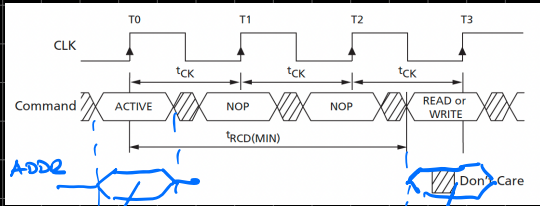
CL → čas potreben, da se informacija iz rep. vrstice v vrstice zapiše (skri I/O podatki elektronsko) v izhodni rep. vrstice, od koder jo poskušamo na volilo

→ to je čas od izstavitve ukaza READ/WRITE do trenutka, ko je informacija dostopna na pod. volilu

t_{RP} → čas potreben, da se informacija iz tip. gajracikla zapiše nazaj v vrstice v banki

Branje:

1. odpiranje vrstice (ACTIVE ukaza)

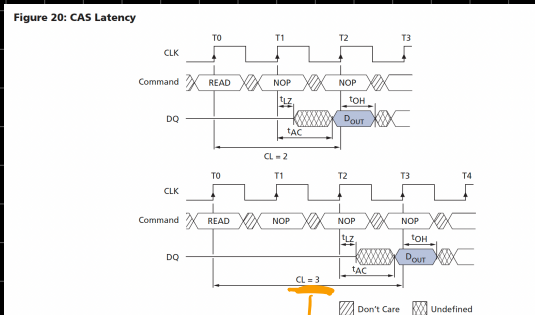


naslov vrstice

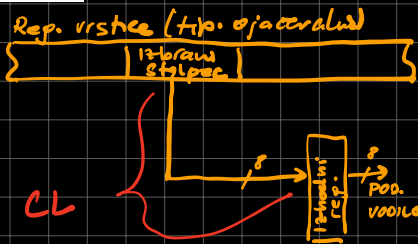
naslov stolpca

→ skupaj z ACTIVE ukazom mora memory controller na naslednje vodila poslati naslov vrstice in koder in nato podatki 2(3) urne periode

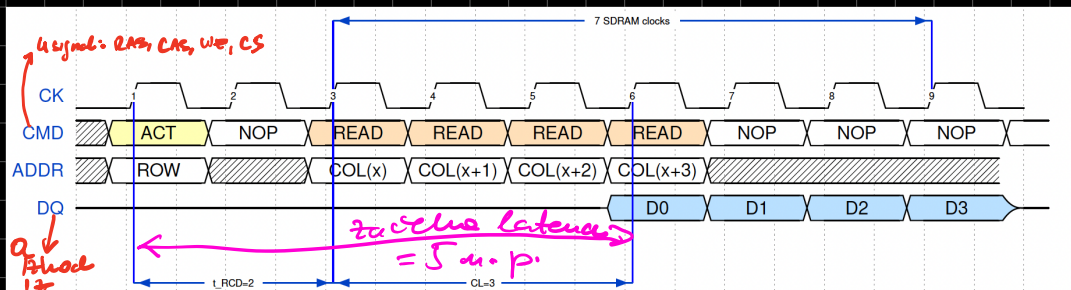
2. READ ukaz (naslov stolpca + READ)



→ ob ukazu READ mora memory controller na naslednje vodila poslati naslov stolpca in podatki CL časa (2-3 u.p.) preden prebere podatke



3. Branje 4 zaporednih stolpcev iz iste vrstice



12-bits stolpca
12-bits stolpca

↙
v 9 urinih periodah dobimo
v 4 zahtevne besede (stolpe)

⇓
 $t_{acc} = 2.25 \text{ u.p.} / \text{beseda}$

V realnem rač. sistemu beremo SDRAM
ob sprejetju v PP → taktot beremo
cel blok !

↙
Intel: 64 B dolina bloka

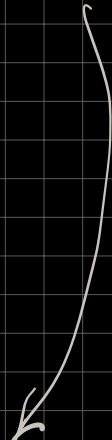
↙
ob vsaki sprejetju beremo 64
stolpcev iz iste vrstice

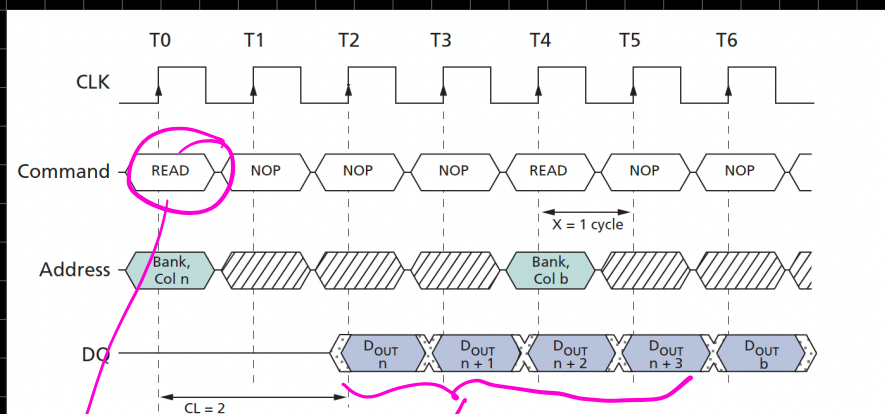
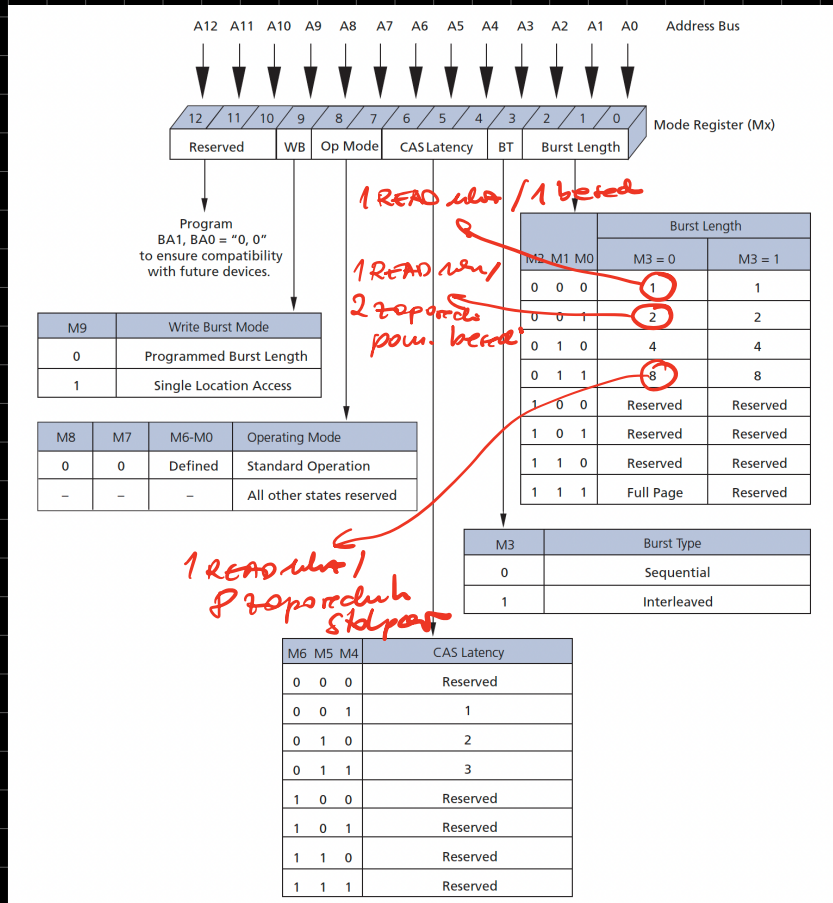
↙
celoten čas = 5 u.p. + 64 u.p. = 69 u.p.

$$t_{acc} = \frac{69}{64} \text{ u.p.} / \text{beseda}$$

* Inicializacija SDRAM-a

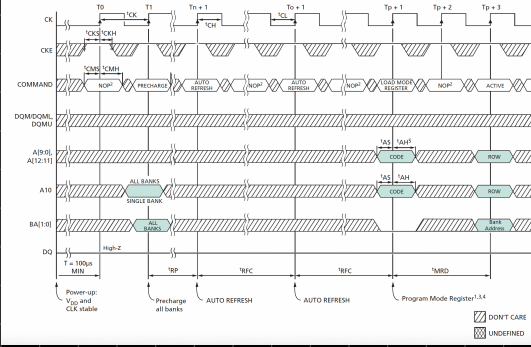
→ pred prvo uporabo SDRAM čipa
(po resetu sistema) moram SDRAM
čip ponastaviti z pisanjem ustreznih
vrednosti v MODE register:





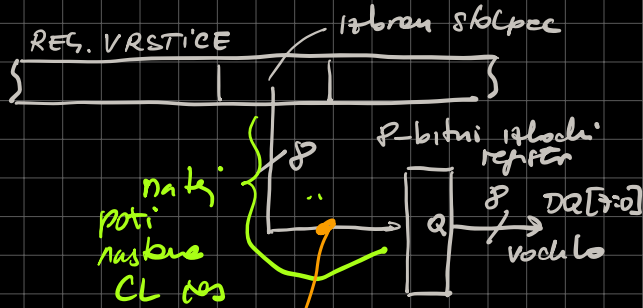
Int sekvenca:

Figure 18: Initialize and Load Mode Register

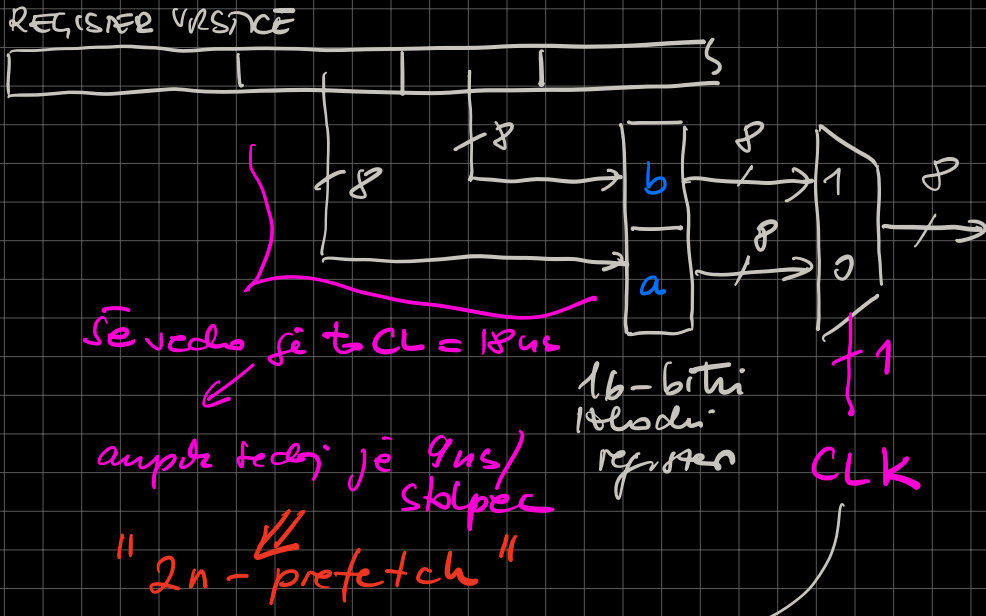


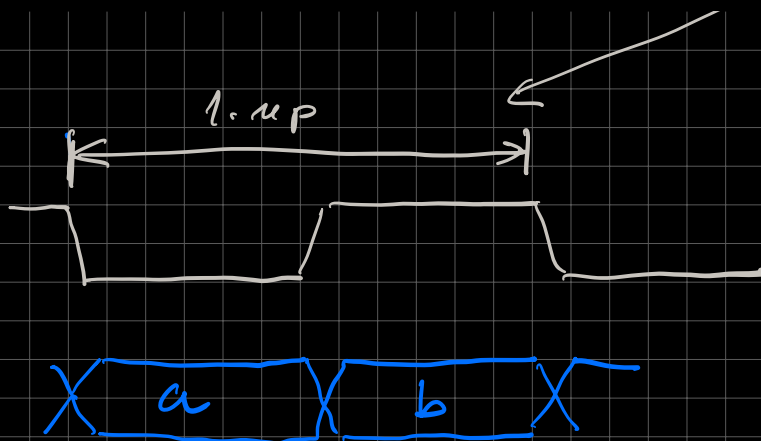
kako pobiti dati dostop?

a) kako "navidezno" skrajovati CL
 000 = maskirati CL 000



Register: 8-bitni register in 8-bitni register





↓
 znotraj 1 urine periode
 bomo iz iste vrstice na
 vodila postali 2 sosednji
 stolpci

⇓
 😊 Podvojili smo
 hitrost prenosa na
 vodila

⇓
 DDR SDRAM

↓
 "Double Data Rate"

Zpored:

če želimo prebrati 64 B,
 oz. 64 zaporednih stolpcev
 iz iste vrstice bomo
 potrebovali:

$$5 \text{ u.p.} + 32 \text{ u.p.} = 37 \text{ u.p.}$$

↓
 za dodatno
 latenco

$$\frac{37 \text{ u.p.}}{64 \text{ bajtov}} = 0.5 \text{ u.p. / bajtov}$$

b) Kako navidezno skrajšati t_{RPD} čr?

↓
predpr. pravilno vrstice =

prebrano sosednjih vrstic iz
drugih bank

Kako se polimira?

DDR2 SDRAM $\Rightarrow$ 4n prefetch

↓

podvojili so frekv
na pod. vodila

DDR3 SDRAM $\Rightarrow$ 8n prefetch

podvojili so frekv
na pod. vodila
v primeru z DDR2

DDR4 SDRAM $\Rightarrow$ 8n prefetch
 $\rightarrow$ podvoje. so frekv

DDR5 SDRAM $\Rightarrow$ 16n prefetch
 $\rightarrow$ podvojeno frekv

77 **Memory Transfers**

Generation		Chip		Data bus		Timings	
DRAM	DRAM name	Clock (Mhz)	Prefetch	Clock (MHz)	MT/s	CL-t _{RCD} -t _{RP}	t _{CL} (ns)
DDR	DDR-266	133		133	266	2.5-3-3	18.8
DDR	DDR-300	150	2N	150	300		
DDR	DDR-400	200	$\rightarrow T_{cu} = 5ns$		400	3-3-3	15
DDR2	DDR2-533	133		266	533	4-4-4	
DDR2	DDR2-667	166	4N	333	667	5-5-5	15
DDR2	DDR2-800	200		400	800	6-6-6	
DDR3	DDR3-1066	133		2x $\rightarrow$ 533	1066	7-7-7	13.12
DDR3	DDR3-1333	166	8N	666	1333	9-9-9	13.5
DDR3	DDR3-1600	200		800	1600	11-11-11	13.75
DDR4	DDR4-2400	300	8N	1200	2400	18-18-18	13.5
DDR4	DDR4-2666	333		1333	2666	20-20-20	13.6
DDR4	DDR4-3200	400	$\rightarrow 4x$		1600	22-22-22	13.75

DDR5

16N