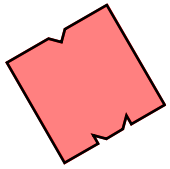
CS-590.26, Spring 2014

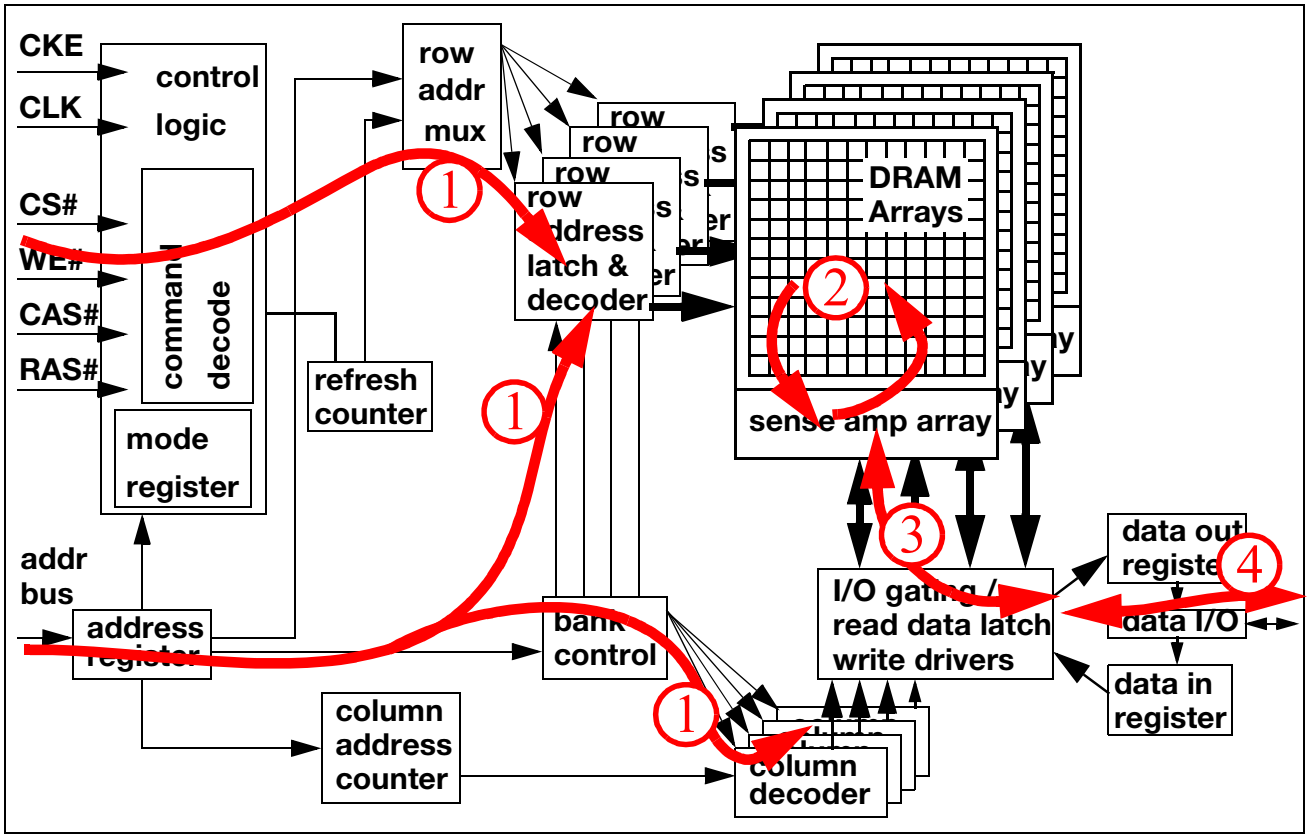
High Speed Memory Systems: Architecture and Performance Analysis

Memory Access Protocol and Memory Controller Basics

Credit where credit is due:
Slides contain original artwork (© Jacob, Wang 2005)



Move Data on and off



1 command transport and decode

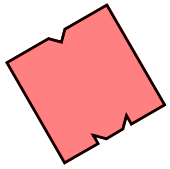
2 in bank data movement

3 in device data movement

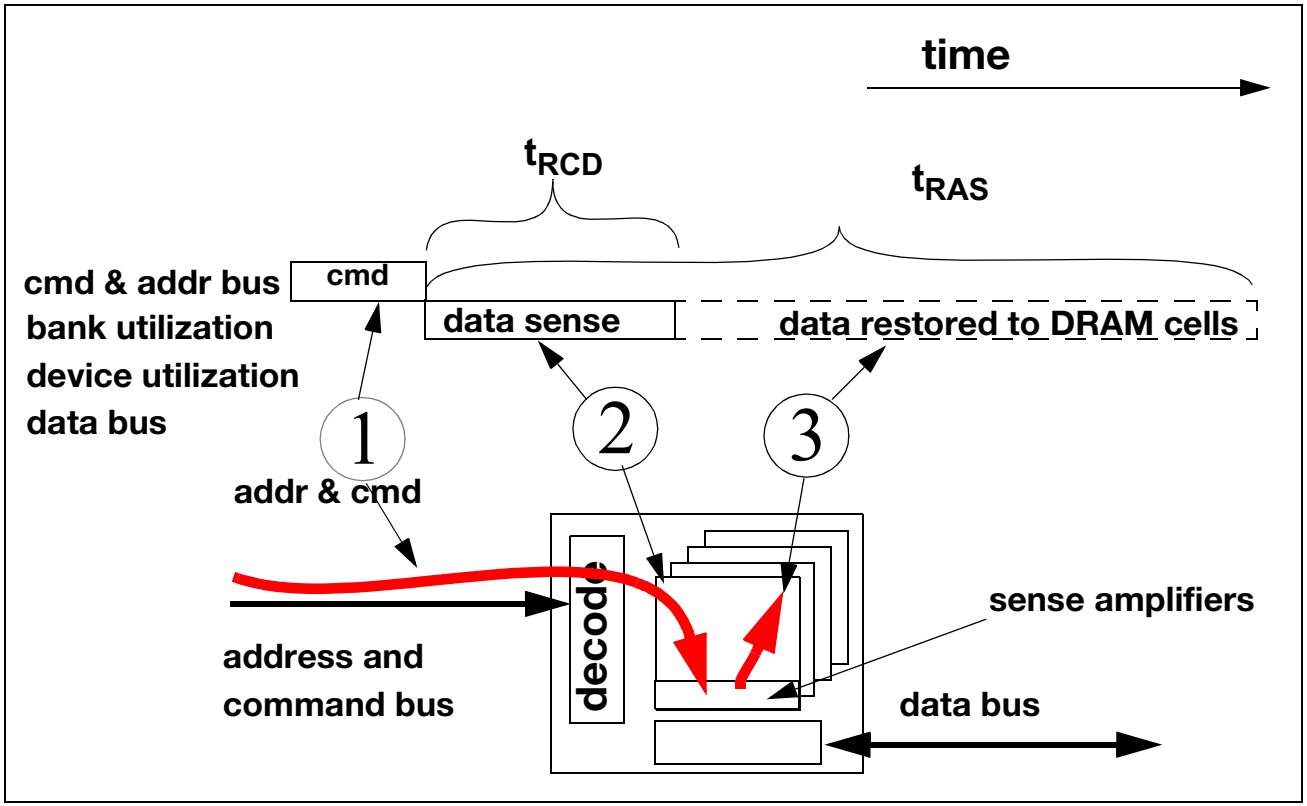
4 system data transport

Basic Commands

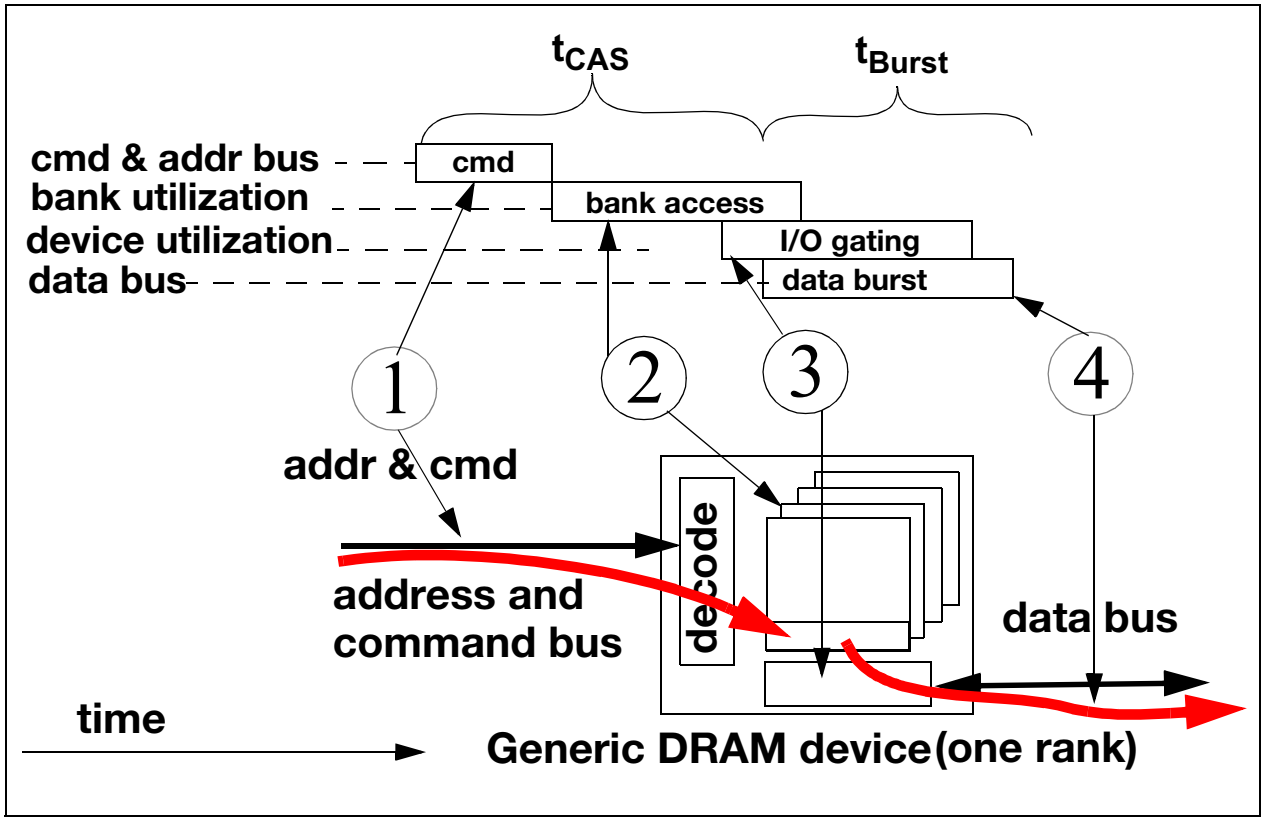
- **Row Access**
- **Column Read**
- **Column Write**
- **Precharge**
- **Refresh**
- **Need other commands to manage special structures. (i.e. Write buffers, virtual channels)**



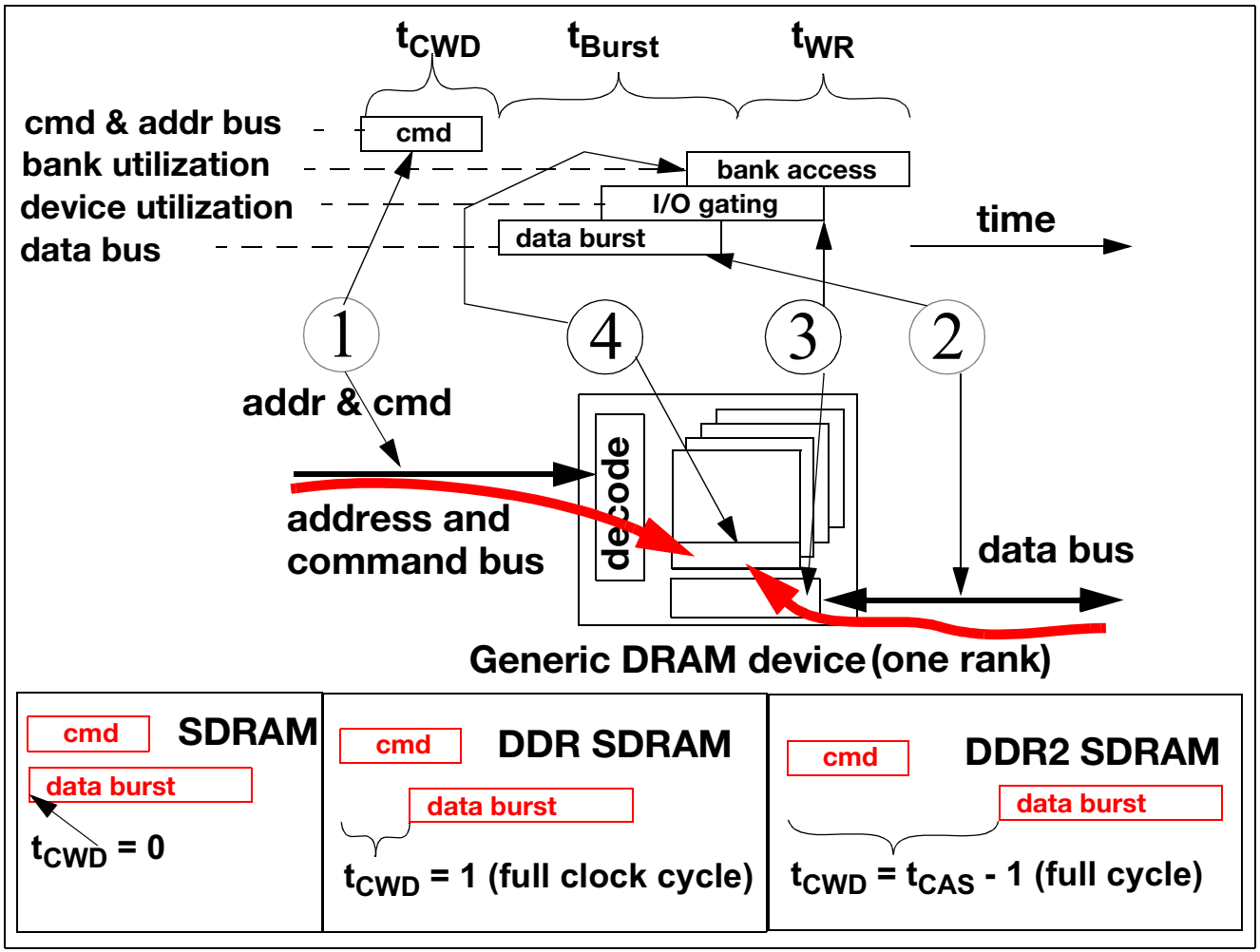
Row Access


time
t_{RCD}
t_{RAS}
cmd & addr bus
bank utilization
device utilization
data bus
cmd
data sense
data restored to DRAM cells
1
2
3
addr & cmd
decode
sense amplifiers
address and
command bus
data bus

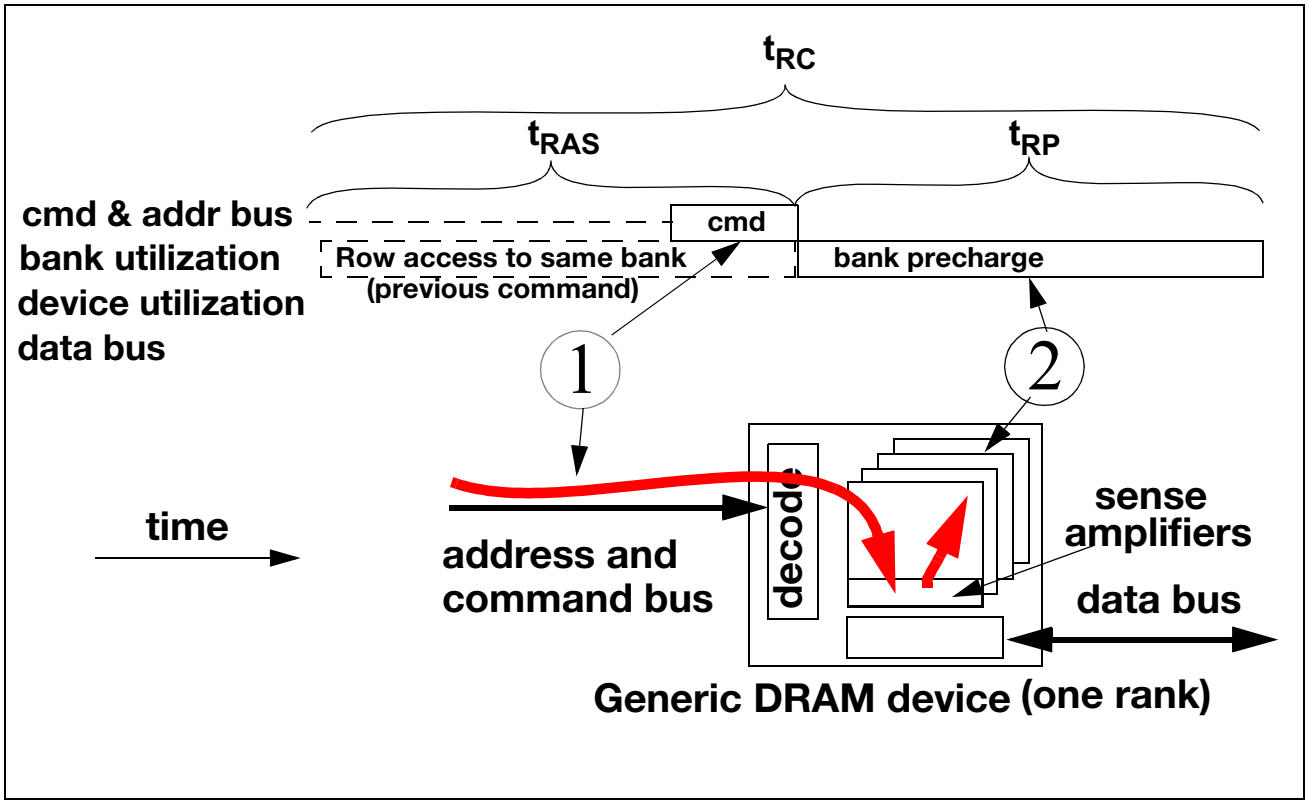
Column Read


t_{CAS}
t_{Burst}
cmd & addr bus
bank utilization
device utilization
data bus
cmd
bank access
I/O gating
data burst
1
2
3
4
addr & cmd
address and
command bus
decode
data bus
time
Generic DRAM device(one rank)

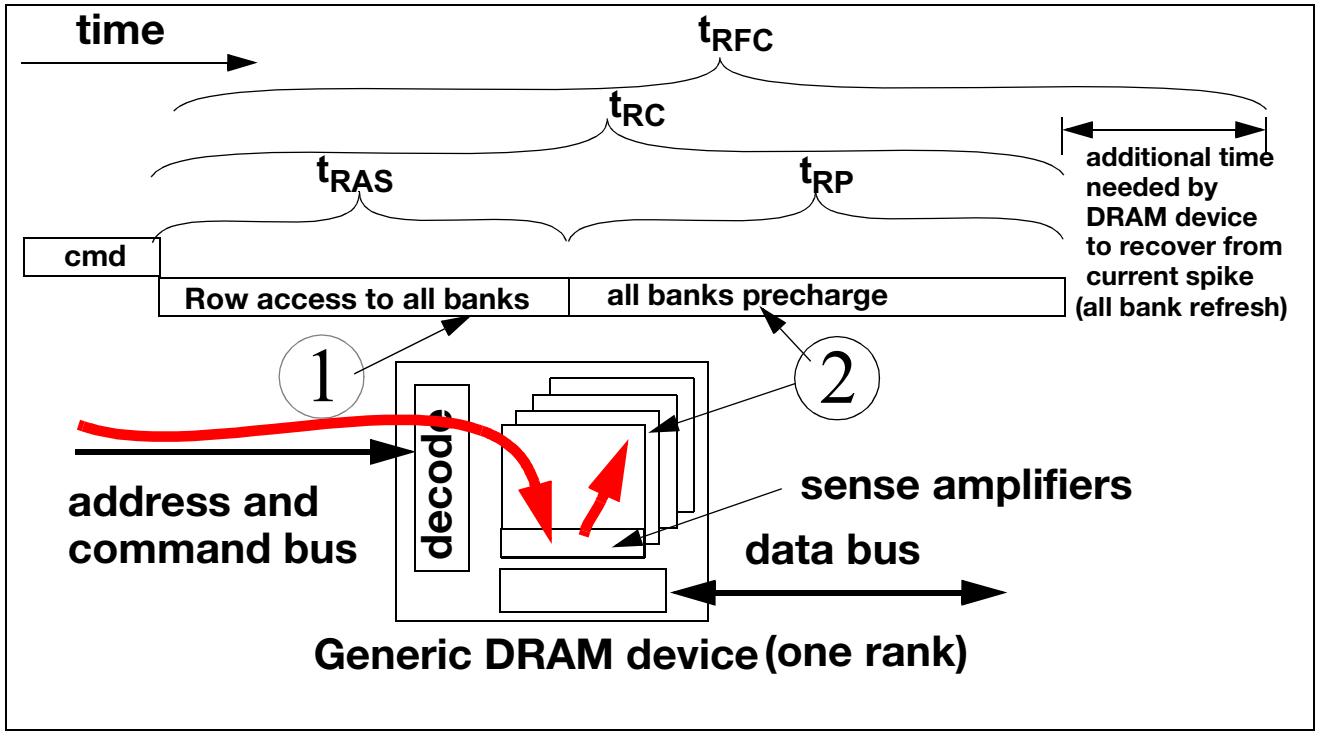
Column Write



Precharge

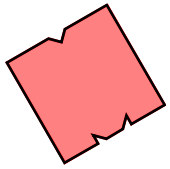

tRC
tRAS
tRP
cmd & addr bus
bank utilization
device utilization
data bus
cmd
Row access to same bank
(previous command)
bank precharge
1
2
time
address and
command bus
decode
sense
amplifiers
data bus
Generic DRAM device (one rank)

Refresh I



Basic Refresh Command: All Banks

* Refresh cycle time dominated by device power consideration, not resource usage. Model is only for resource usage. In real devices, refresh cycle times take longer than tRC. Getting much wose due to larger rows. Parallel refresh hitting all banks draws large current spike. Takes longer time to recover for higher density devices.

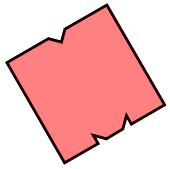


Refresh II

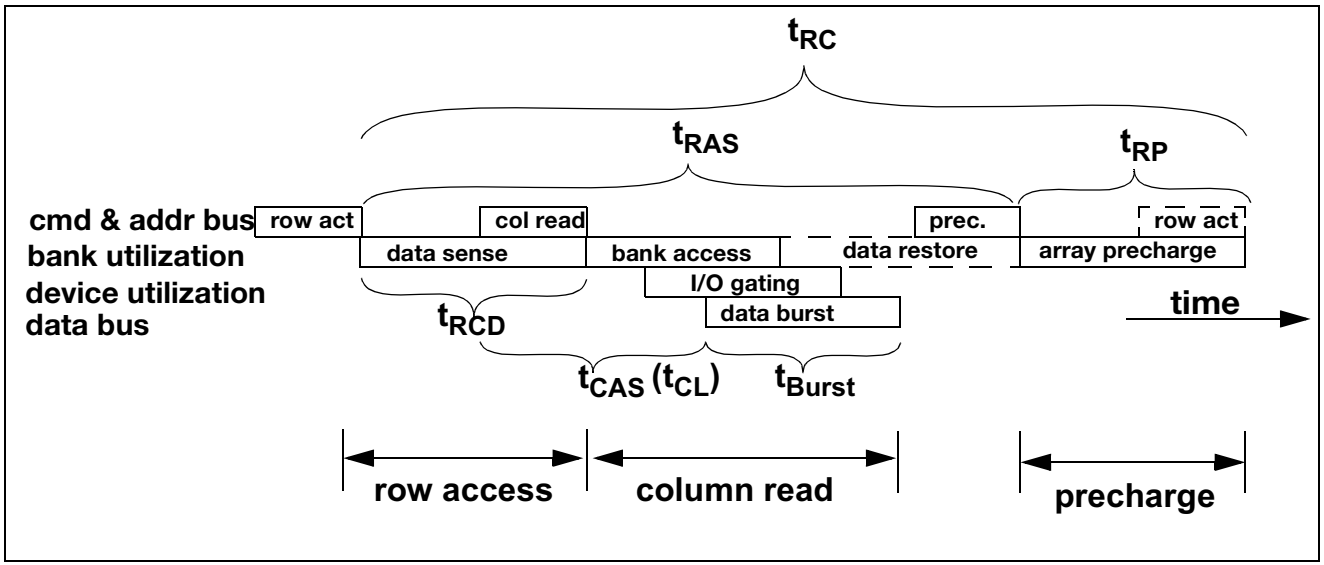
Density	Bank Count	Row Count	Row Size (bits)	Row cycle time (ns)	Refresh Cycle time (ns)
256 Mbit (x8)	4	8192	8192	55	75
512 Mbit (x8)	4	16384	8192	55	105
1 Gbit (x8)	8	16384	8192	55	127.5

Trend: Refresh getting more expensive

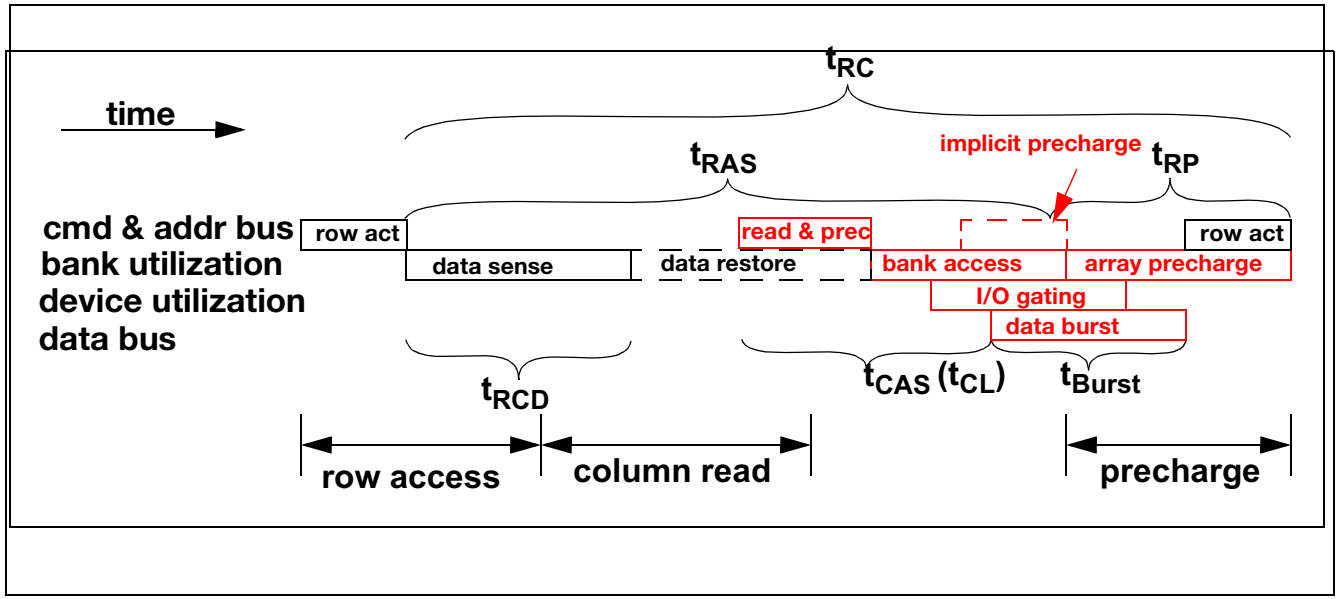
Advanced Refresh? Per Bank, Hidden?

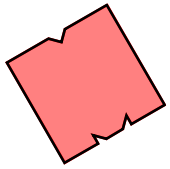


A Read Cycle

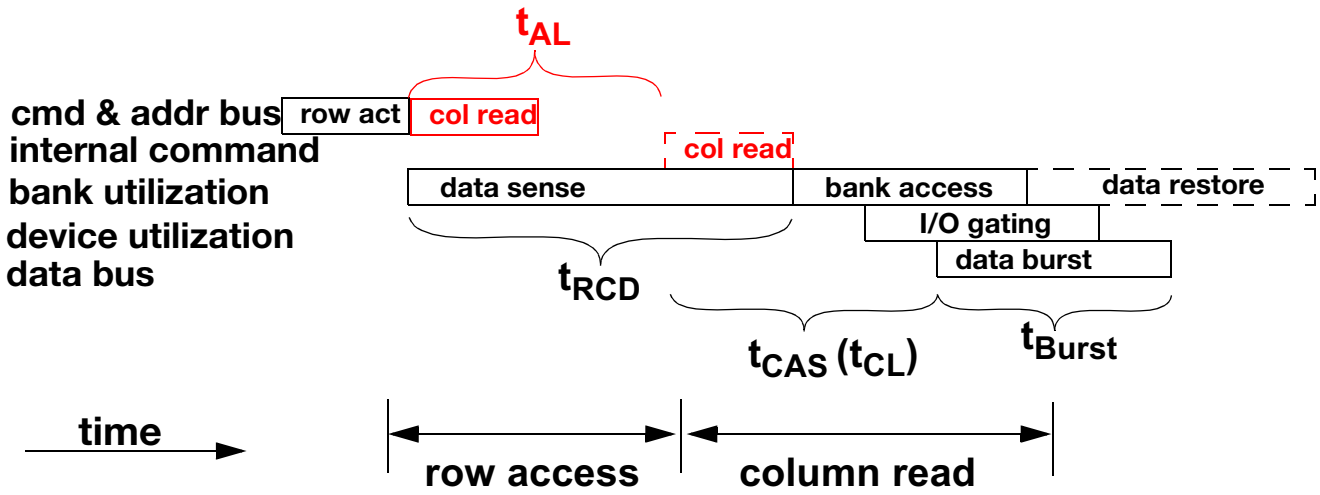

tRC
tRAS
tRP
cmd & addr bus
bank utilization
device utilization
data bus
row act
col read
prec.
row act
data sense
bank access
data restore
array precharge
I/O gating
data burst
time
tRCD
tCAS (tCL)
tBurst
row access
column read
precharge

Read and Precharge



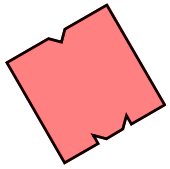


Posted CAS

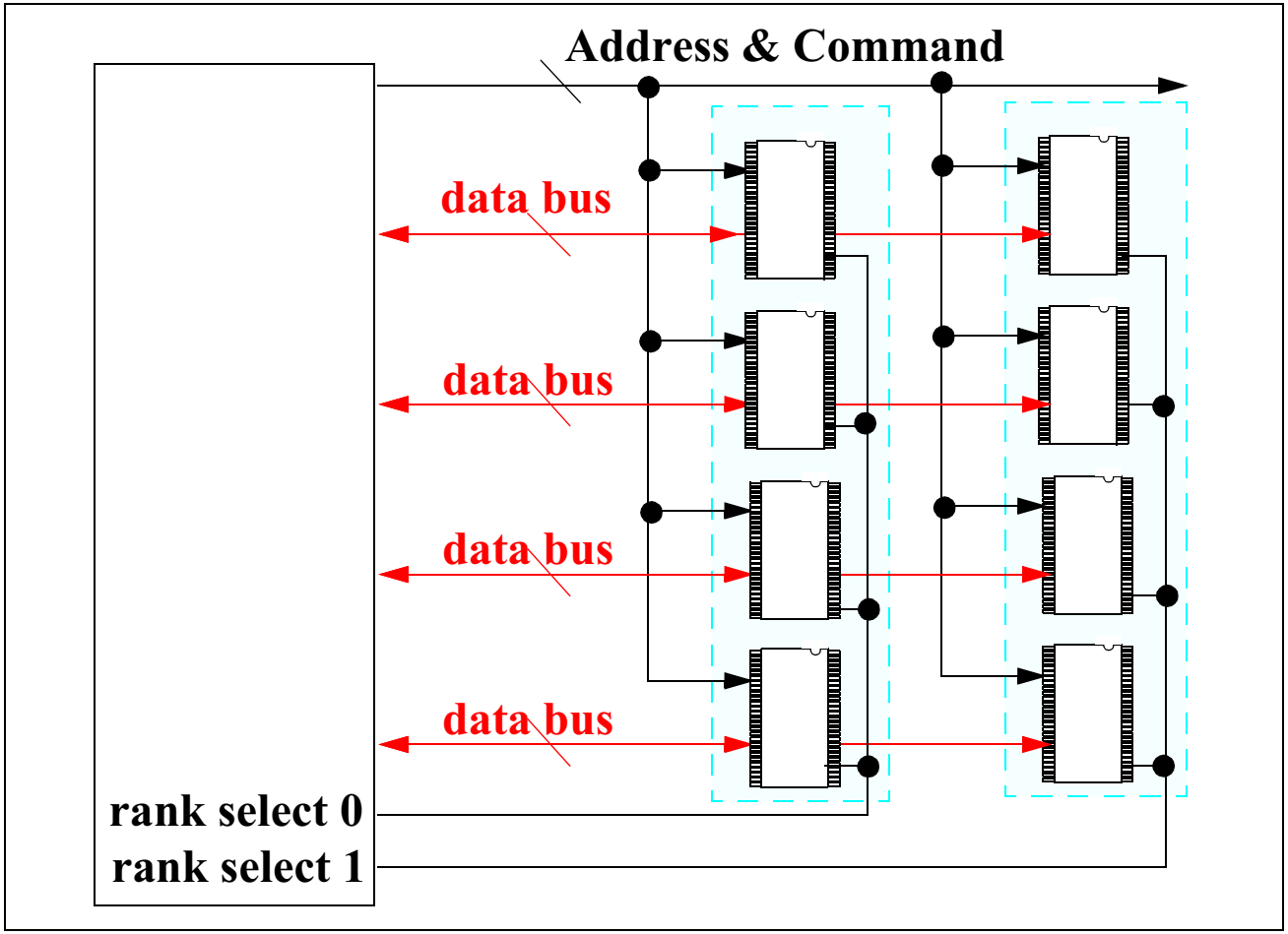


CAS timing remains the same, but delay CAS command internally for X cycles.

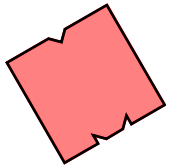
Simplifies controller design.



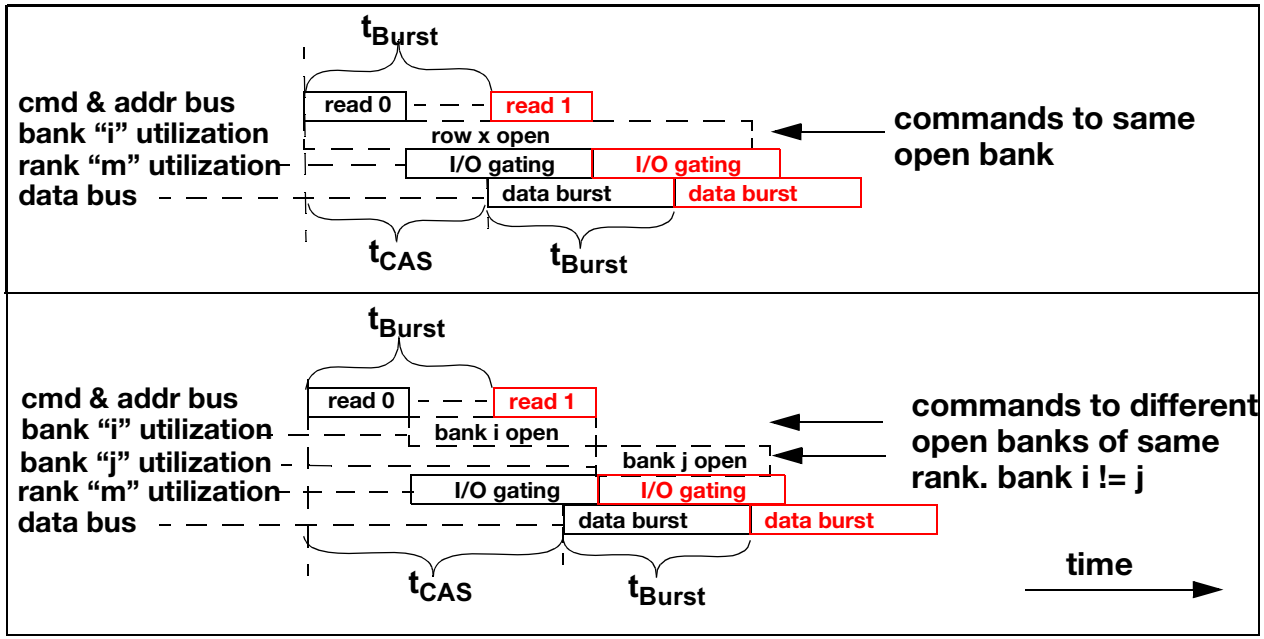
Command Interactions



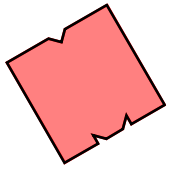
How do DRAM command interactions look like in system context?



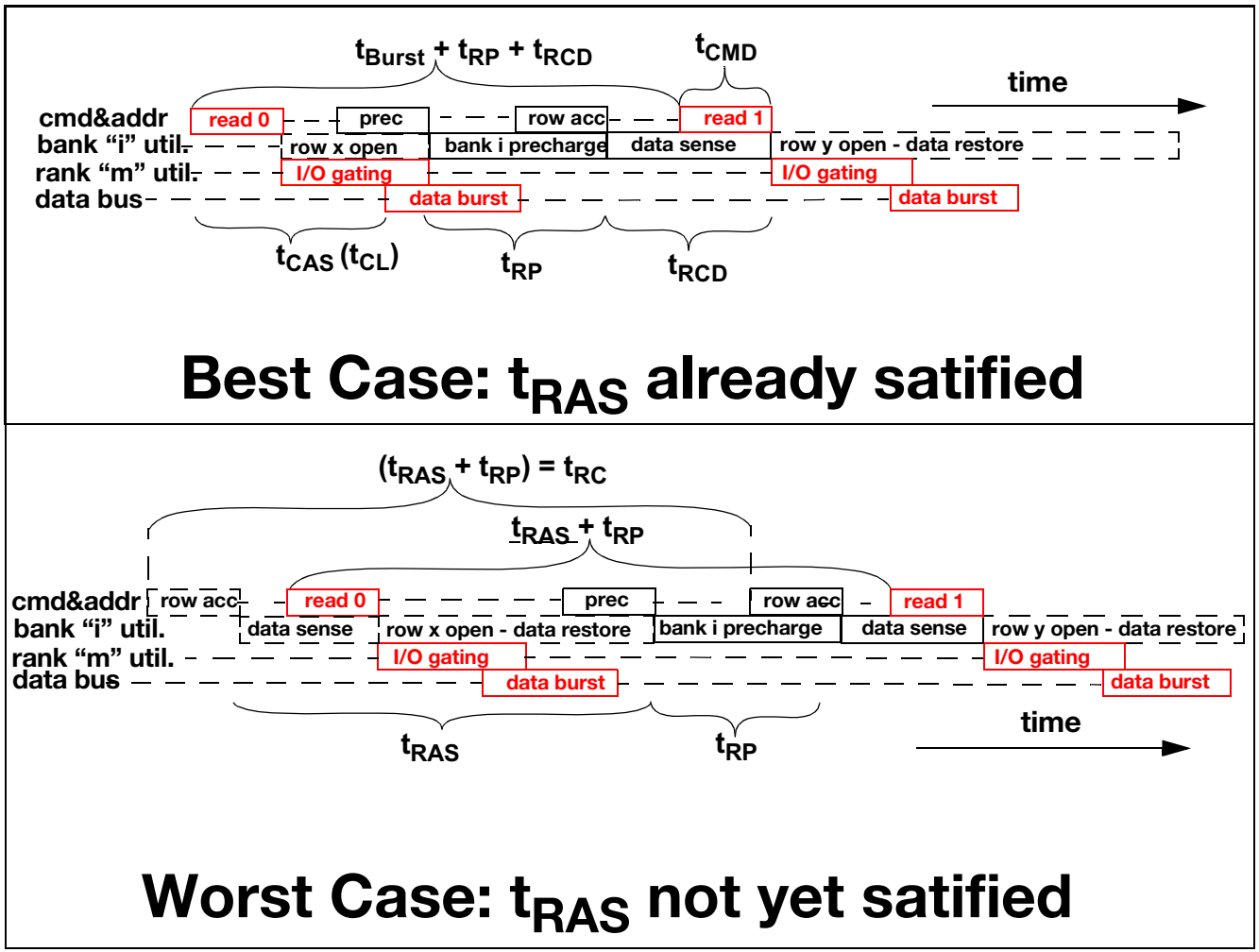
Reads to Same Rank

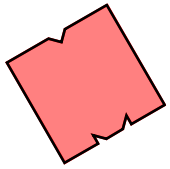


Can be pipelined consecutively in SDRAM/DDRx SDRAM/DRDRAM memory systems



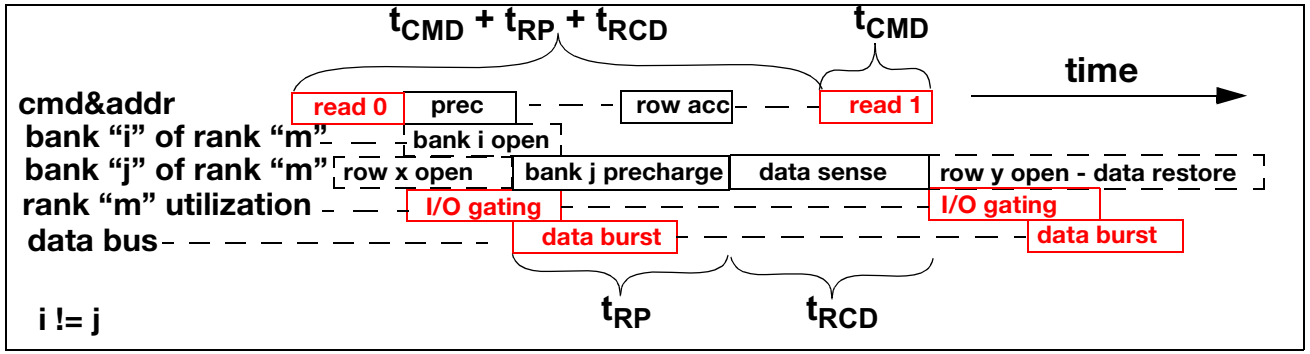
Reads to Diff. Rows of Same Bank





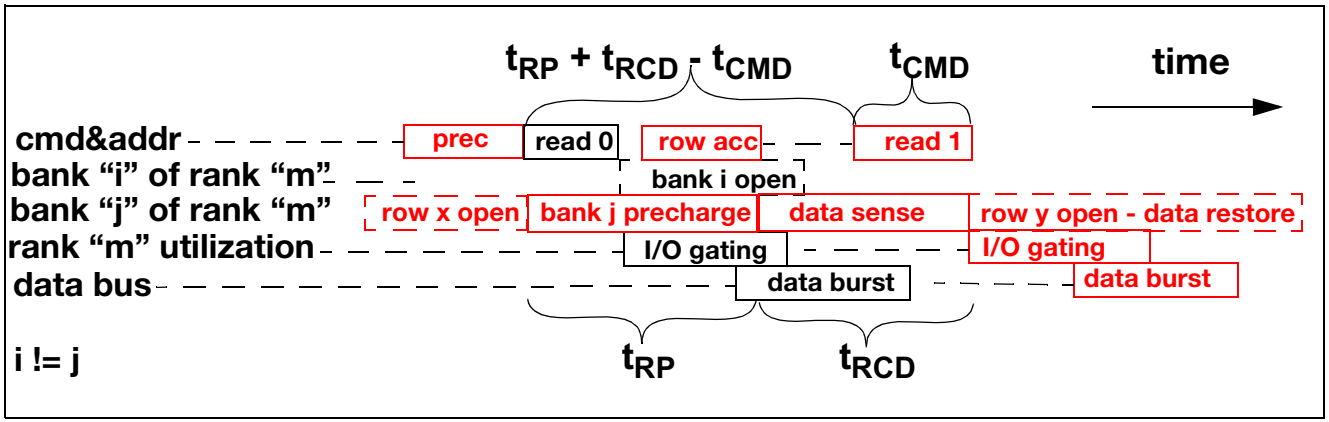
Read to Diff. Banks/Conflict

Two reads, second read to different bank, but has bank conflict.

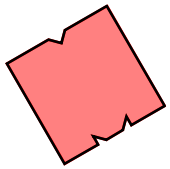


Support Re-ordering?

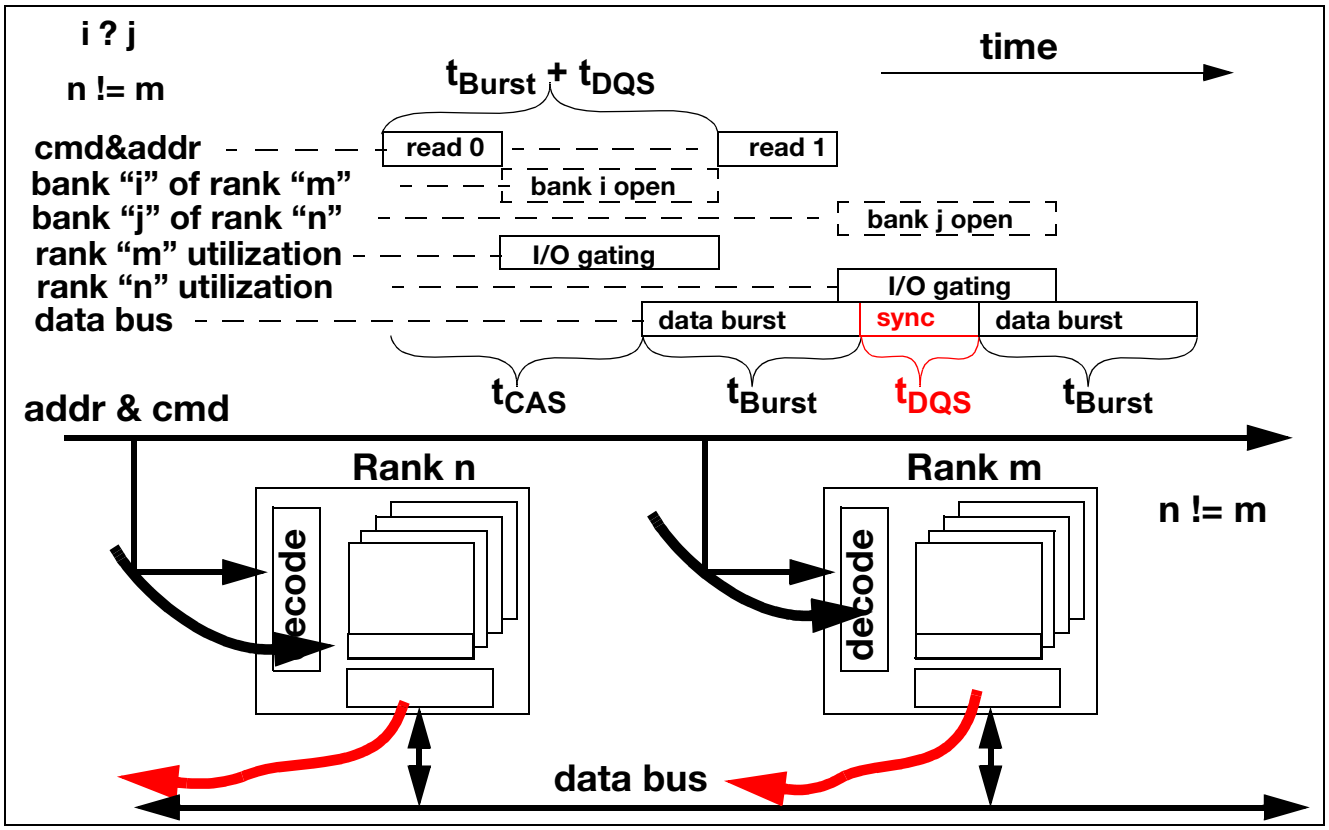
If not... Best case shown above



If yes... Precharge ASAP



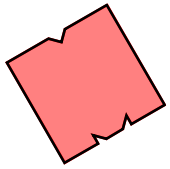
Read to Different Ranks



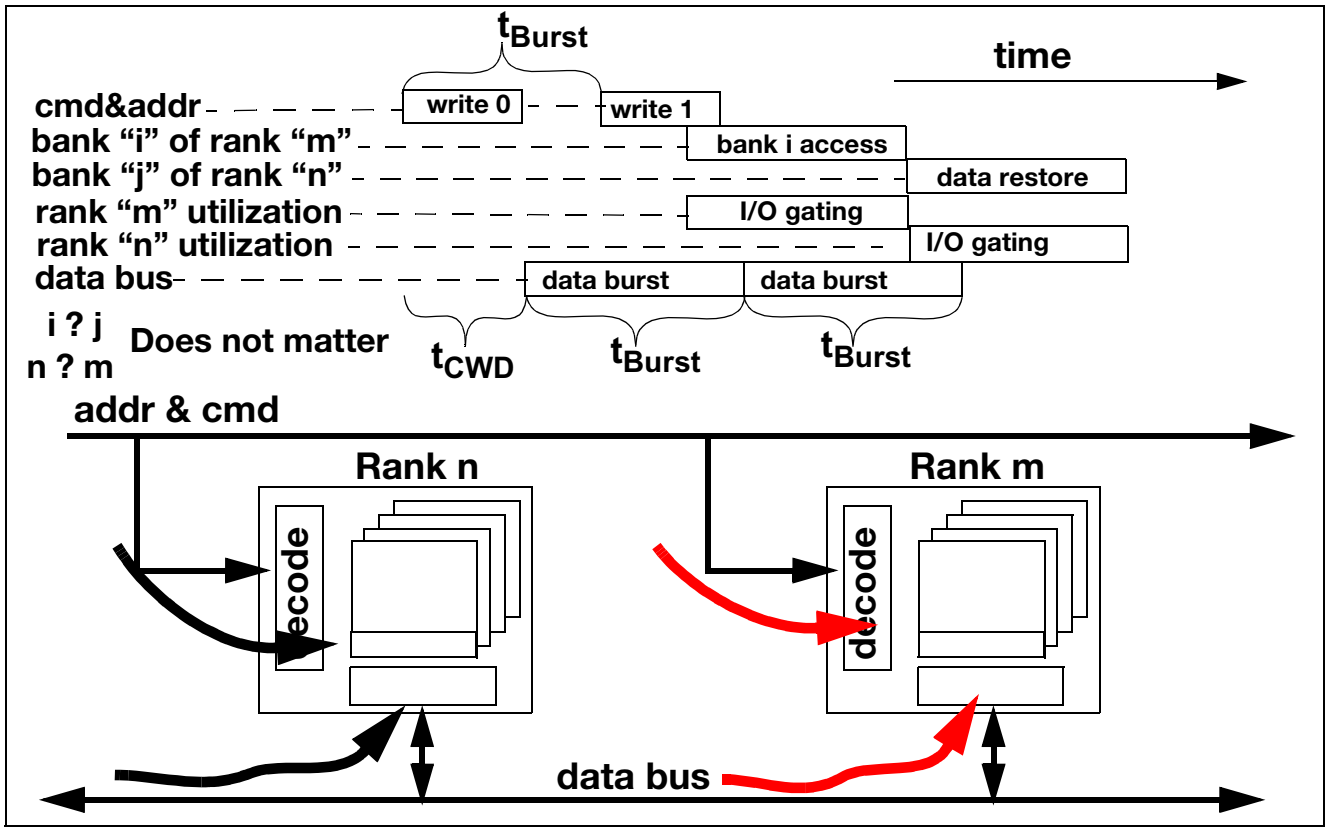
SDRAM: $t_{DQS} = 0$

DDRx SDRAM: t_{DQS} >= 1 full cycle

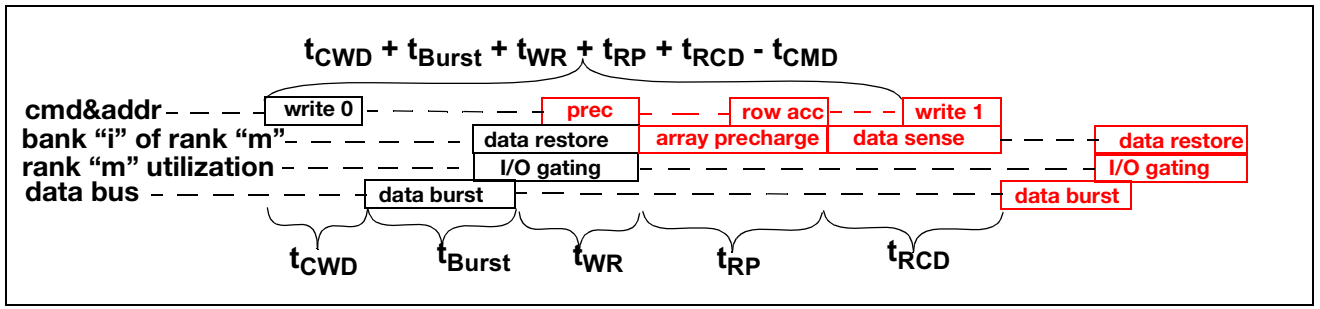
Direct RDRAM: $t_{DQS} = 0$



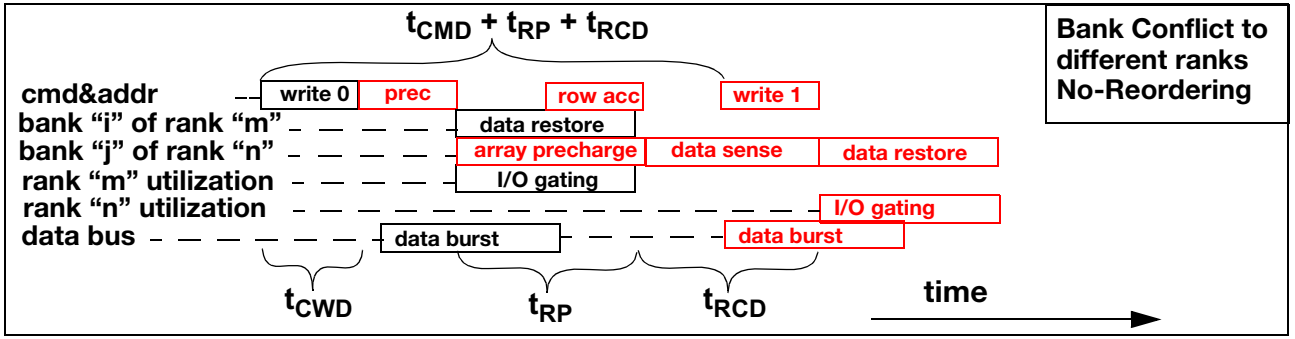
Consecutive Writes



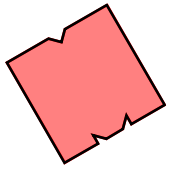
Writes W/Bank Conflict



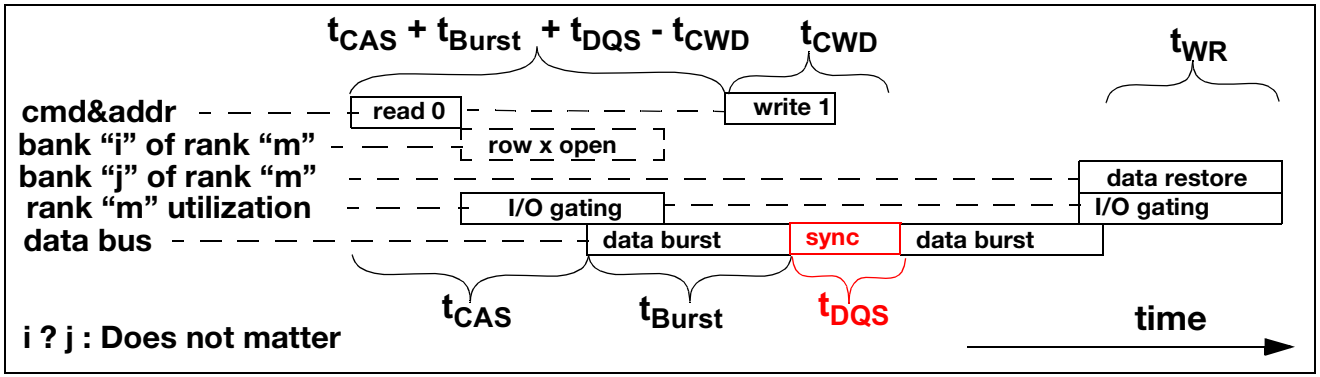
Bank conflict to same bank: best case



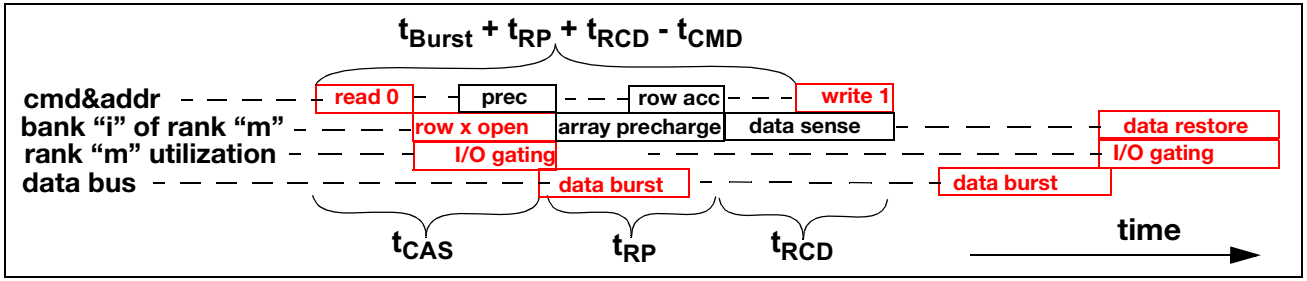
Bank conflict to different ranks: No re-ordering



Write following Read

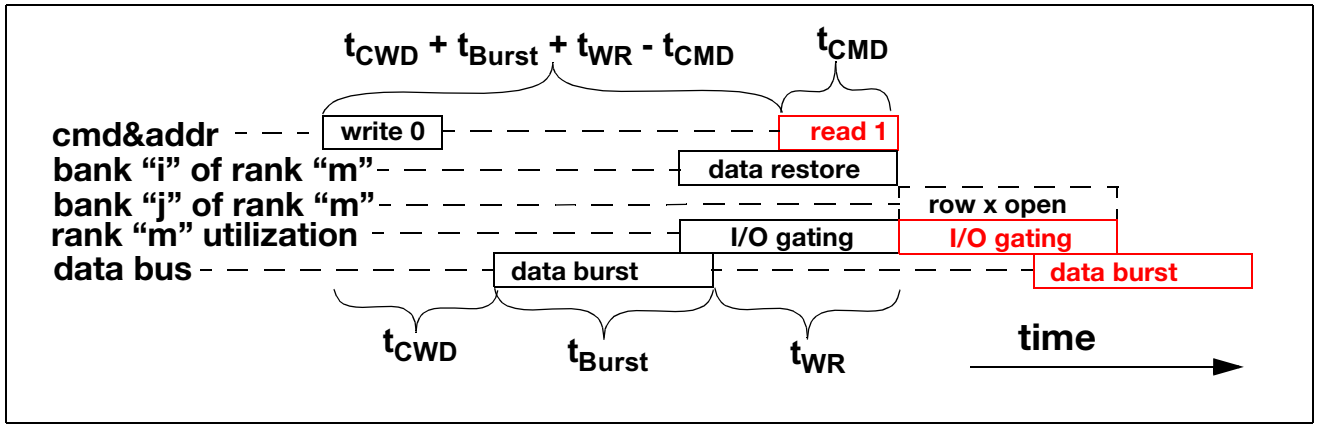


Open Banks: Best Case



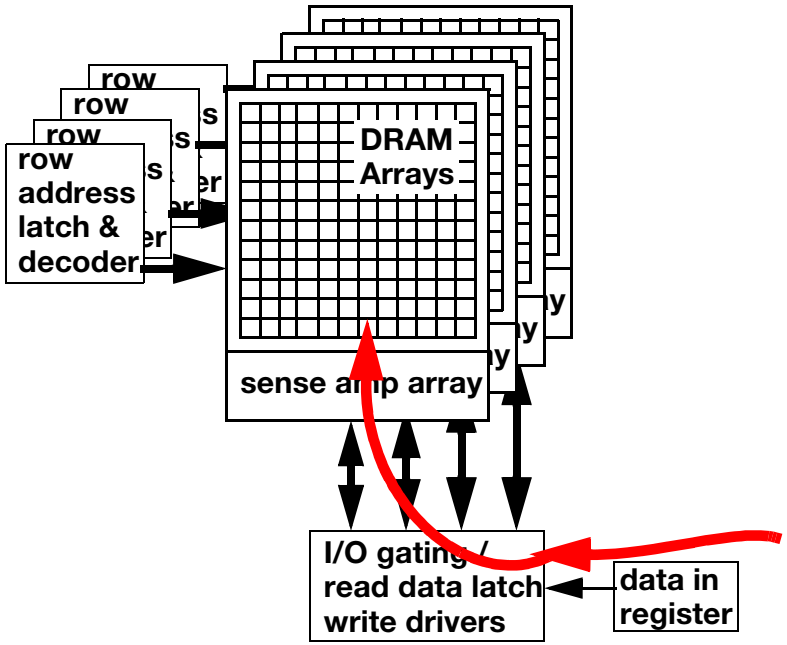
Bank Conflict: Best Case

Read following Write I

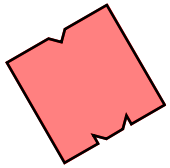


To same rank:

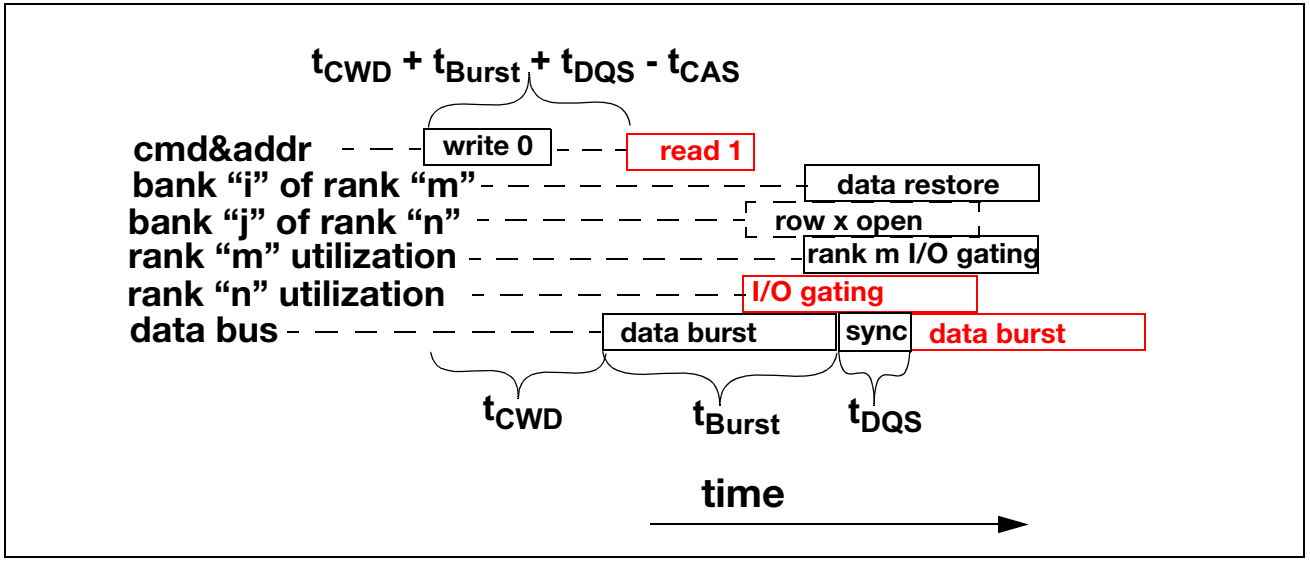
SDRAM/DDRx no write buffer, writes must complete before reads can begin.



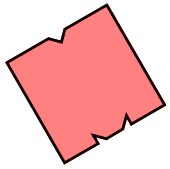
D-RDRAM: has write buffer



Read following Write II

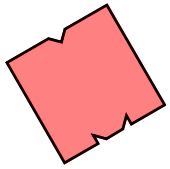


To different ranks: read can proceed "immediately", subject to data bus sharing constraint.

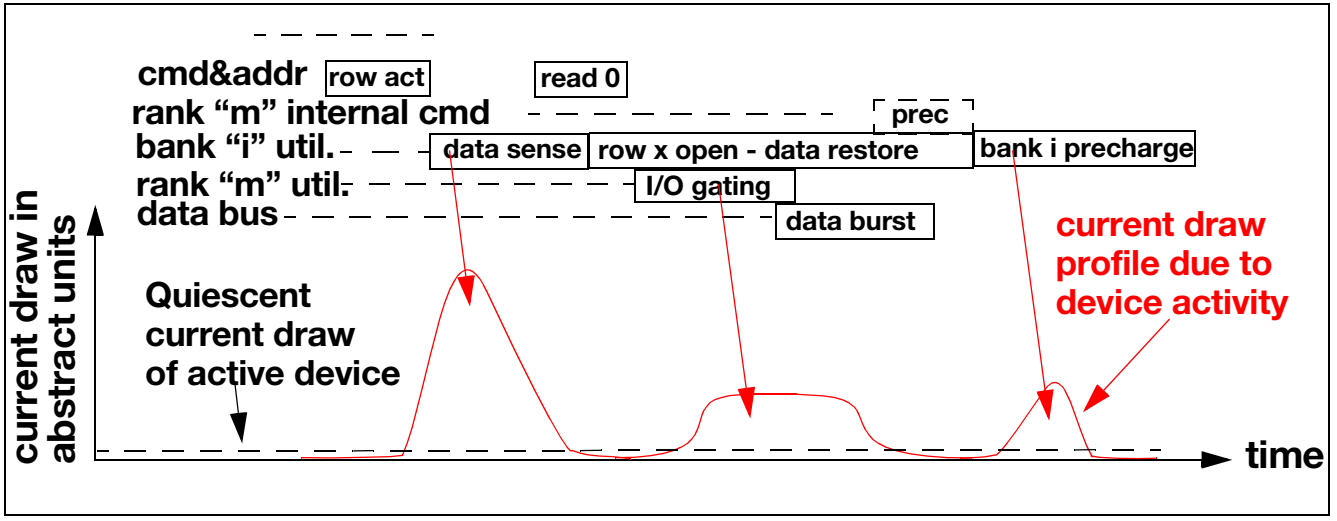


Minimum Command Distances

previous	next	rank	bank	row	Minimum scheduling distance between DRAM commands Open Page No Command Re-Ordering Best Case	Minimum scheduling distance between DRAM commands Worst Case
R	R	s	s	o	t_{Burst}	-
R	R	s	s	c	$t_{Burst} + t_{RP} + t_{RCD}$	t_{RC}
R	R	s	d	o	t_{Burst}	-
R	R	s	d	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
R	R	d	-	o	$t_{DQS} + t_{Burst}$	-
R	R	d	-	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
R	W	s	s	o	$t_{CAS} + t_{Burst} + t_{DQS} - t_{CWD}$	-
R	W	s	s	c	$t_{Burst} + t_{RP} + t_{RCD} - t_{CWD}$	t_{RC}
R	W	s	d	o	$t_{CAS} + t_{Burst} + t_{DQS} - t_{CWD}$	-
R	W	s	d	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
R	W	d	-	o	$t_{CAS} + t_{Burst} + t_{DQS} - t_{CWD}$	-
R	W	d	-	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
W	R	s	s	o	$t_{CWD} + t_{Burst} + t_{WR} - t_{CMD}$	-
W	R	s	s	c	$t_{CWD} + t_{Burst} + t_{WR} + t_{RP} + t_{RCD} - t_{CMD}$	t_{RC}
W	R	s	d	o	$t_{CWD} + t_{Burst} + t_{WR} - t_{CMD}$	-
W	R	s	d	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
W	R	d	-	o	$t_{CWD} + t_{Burst} + t_{DQS} - t_{CAS}$	-
W	R	d	-	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
W	W	s	s	o	t_{Burst}	-
W	W	s	s	c	$t_{CWD} + t_{Burst} + t_{WR} + t_{RP} + t_{RCD} - t_{CMD}$	t_{RC}
W	W	s	d	o	t_{Burst}	-
W	W	s	d	c	$t_{CWD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$
W	W	d	-	o	t_{Burst}	-
W	W	d	-	c	$t_{CMD} + t_{RP} + t_{RCD}$	$t_{RC} - t_{Burst}$

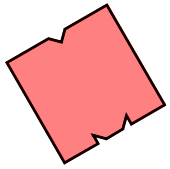


Power Constraint

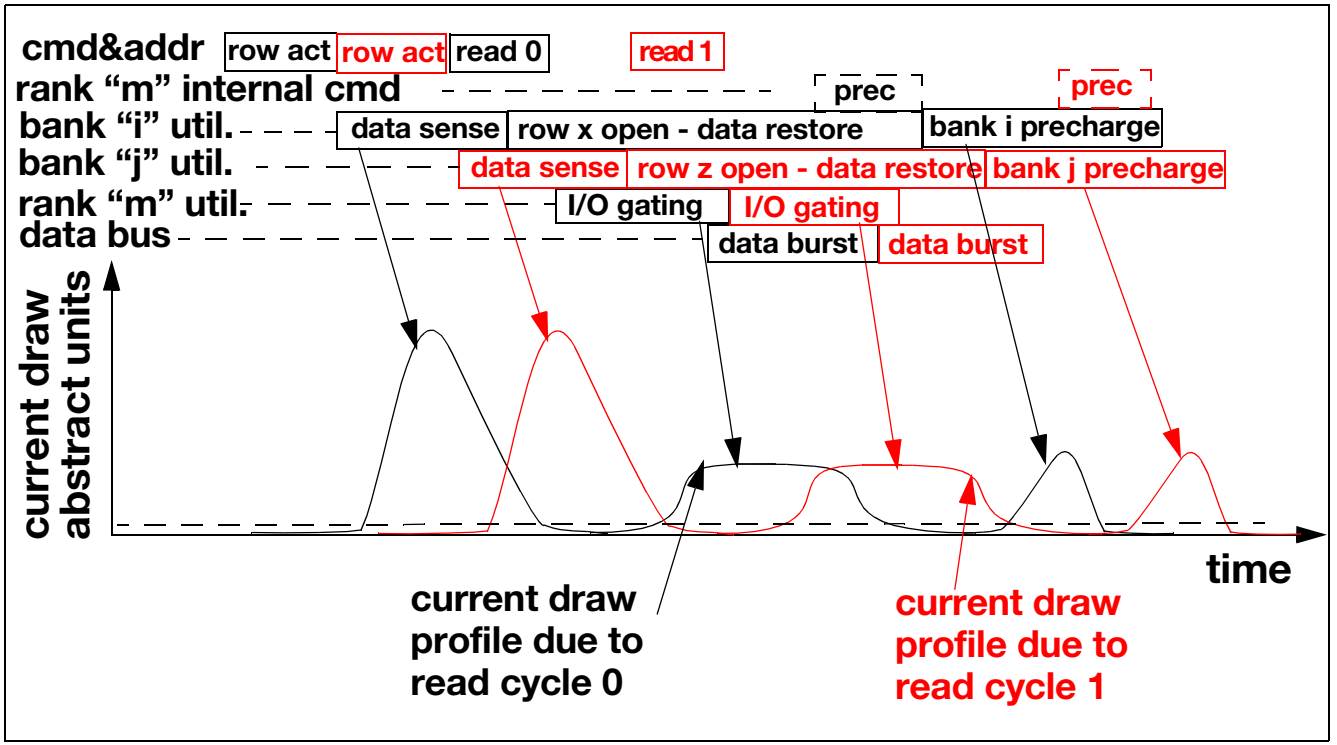


Current profile of a read cycle

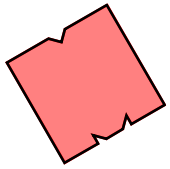
Row Activation draws a lot of current



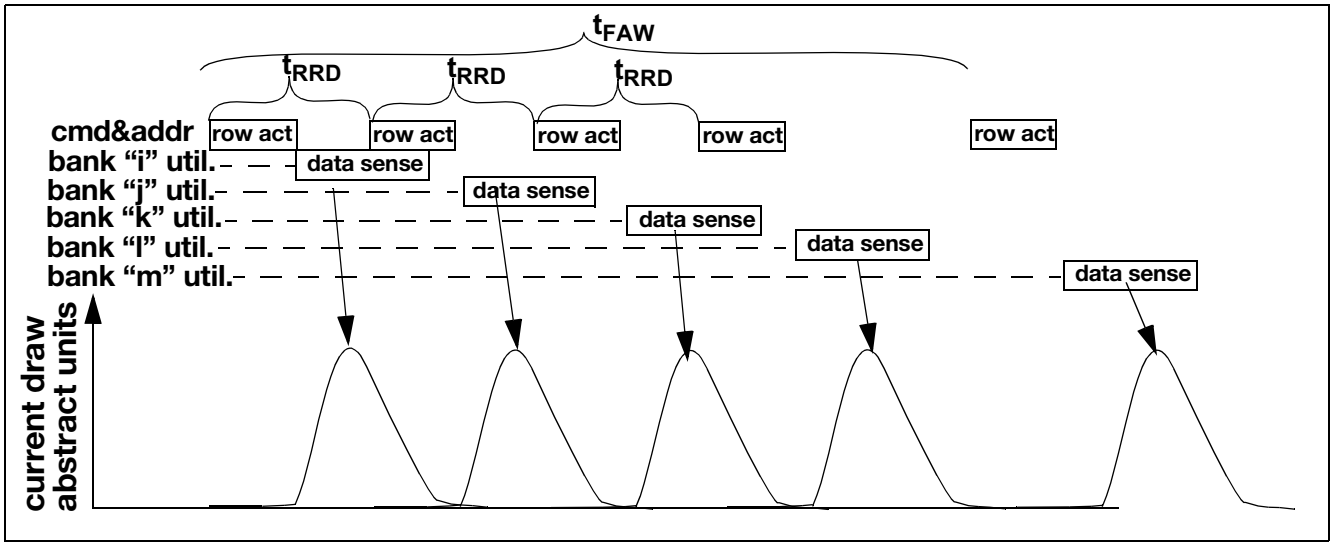
Pipelined Read Cycles



**Start to pipeline DRAM commands,
and it can get really bad**

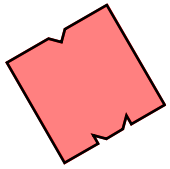


Dumb Solution: t_{RRD} & t_{FAW}



t_{RRD} and t_{FAW} spec'ed to keep row activation commands far apart.

Limits peak current draw, but random row access performance suffers



Worse with Larger Rows

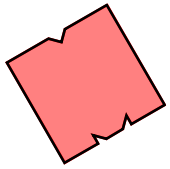
s

Device configuration	512 Mbit x 4	256 Mbit x 8	128 Mbit x 16
Data bus width	4	8	16
Number of banks	8	8	8
Number of rows	16384	16384	8192
Number of columns	2048	1024	1024
Row size (bits)	8192	8192	16384
t_{RRD} (ns)	7.5	7.5	10
t_{FAW} (ns)	37.5	37.5	50

1 Gbit DDR2 SDRAM Devices

Larger Rows = longer t_{RRD} & t_{FAW}
Same issue as t_{RFC}

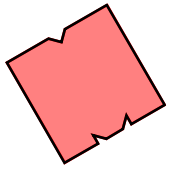
Power consumption IS important for DRAM



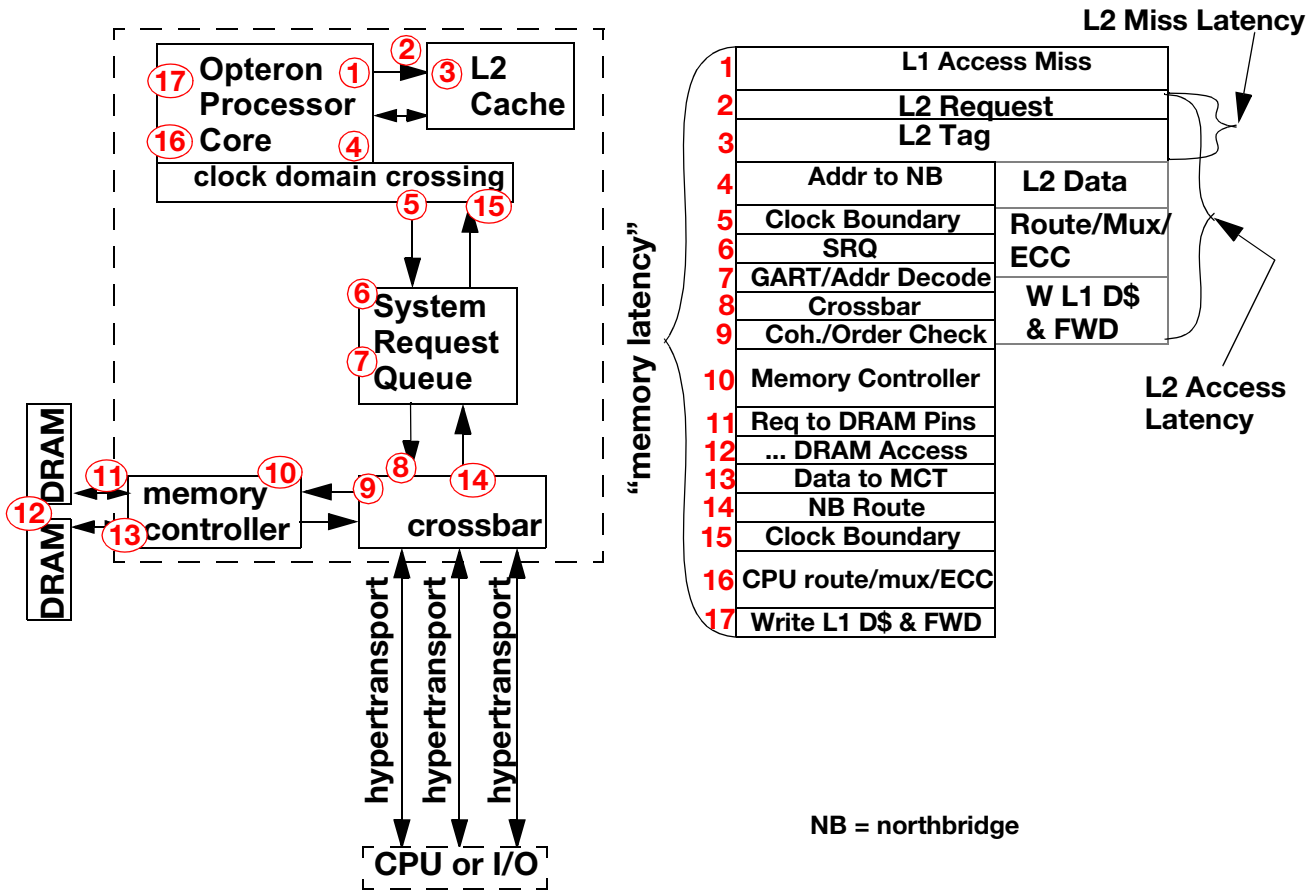
Take a Breath

- More than you ever wanted to know about DRAM command scheduling.
- DRAM commands are quite simple, but command combinations increase complexity.
- Minimum scheduling distances based on resource usage/availability model, but additional constraints exist. i.e. Power limit.
- Additional resources (i.e. write buffers, caches, virtual channels) require additional commands.

Now, on to Controller Basics ...

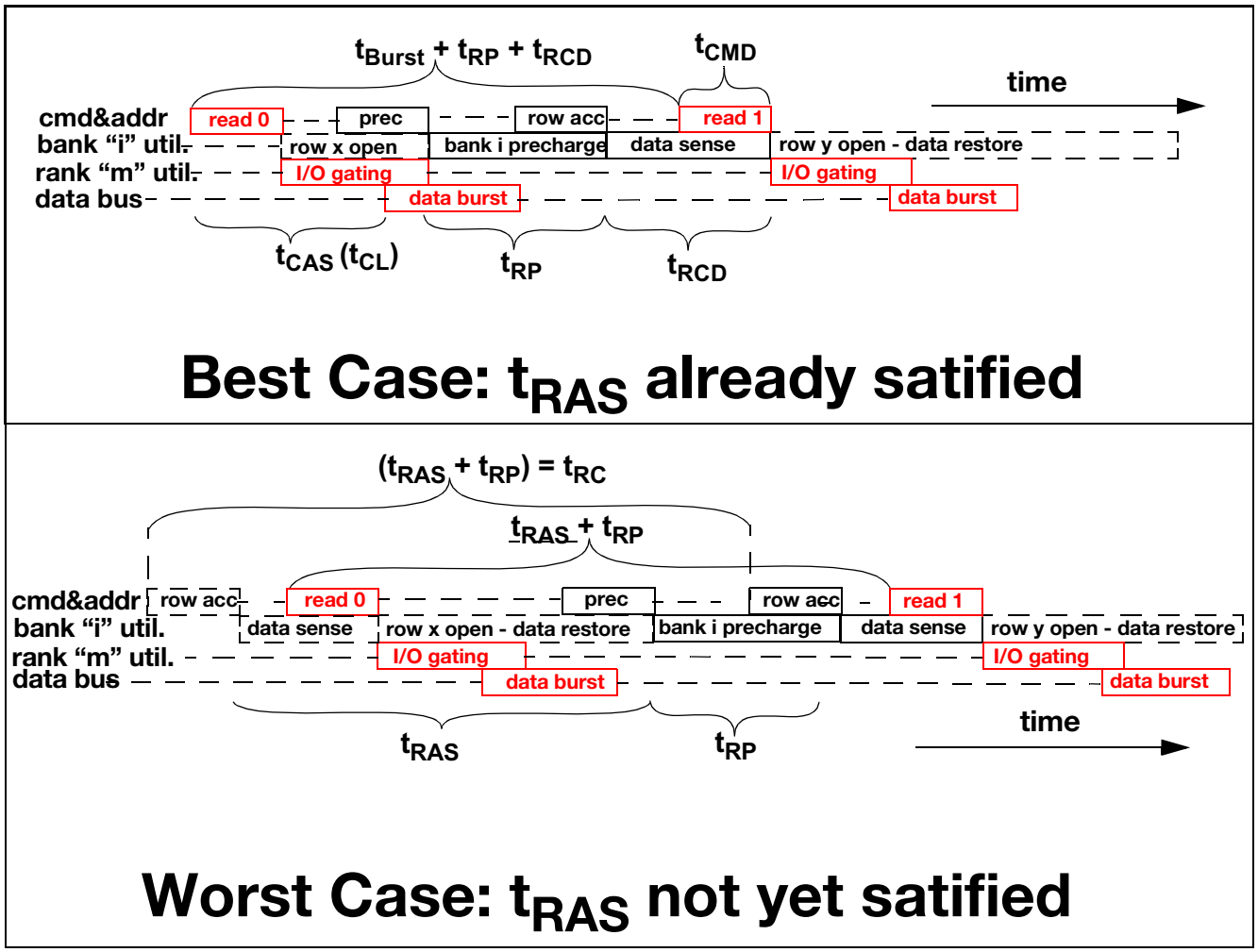


Seen This Before

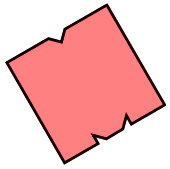


NB = northbridge

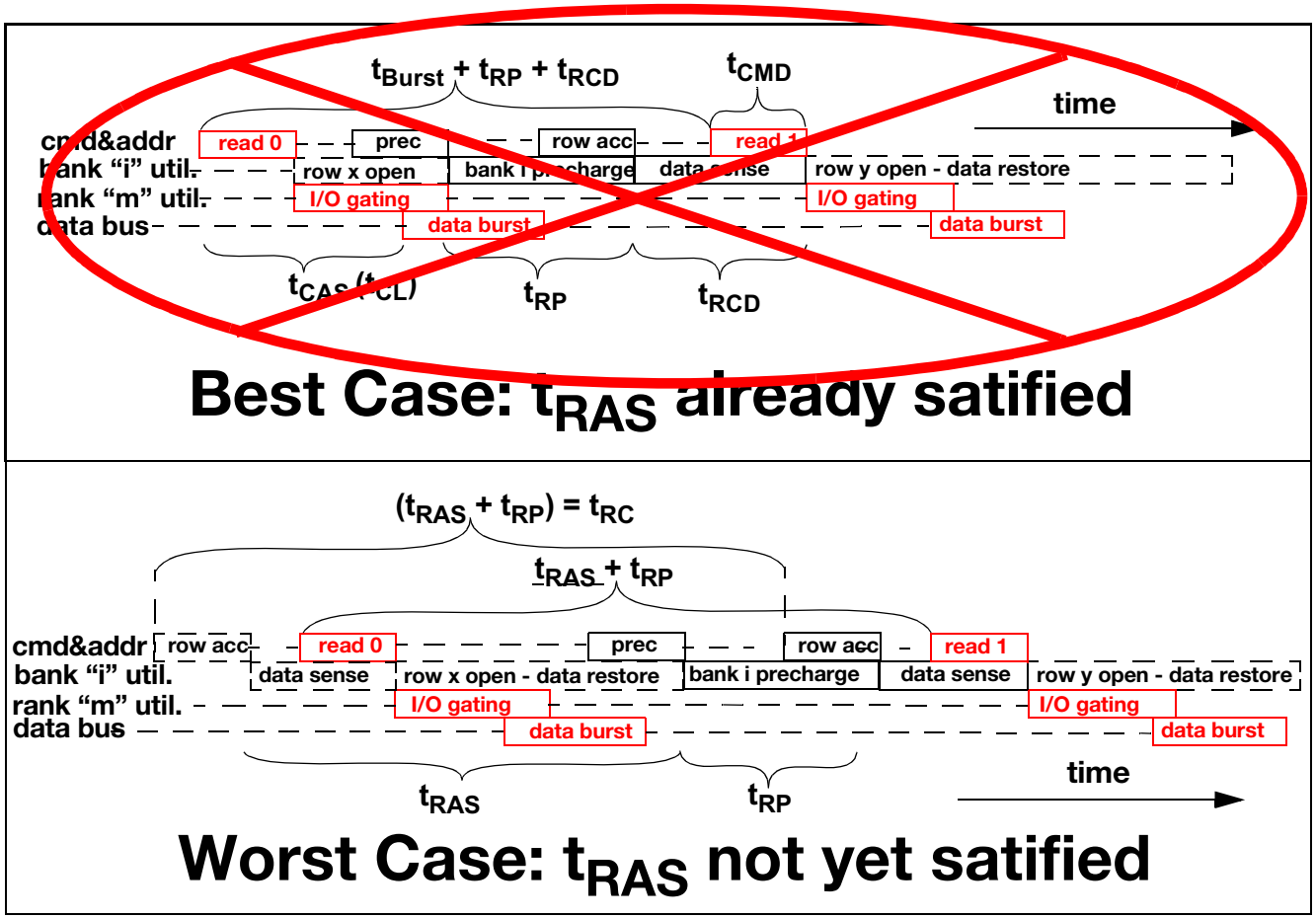
Reads to Diff. Rows of Same Bank



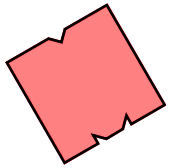
Smart controller can extract maximum performance (lowest latency + max BW)



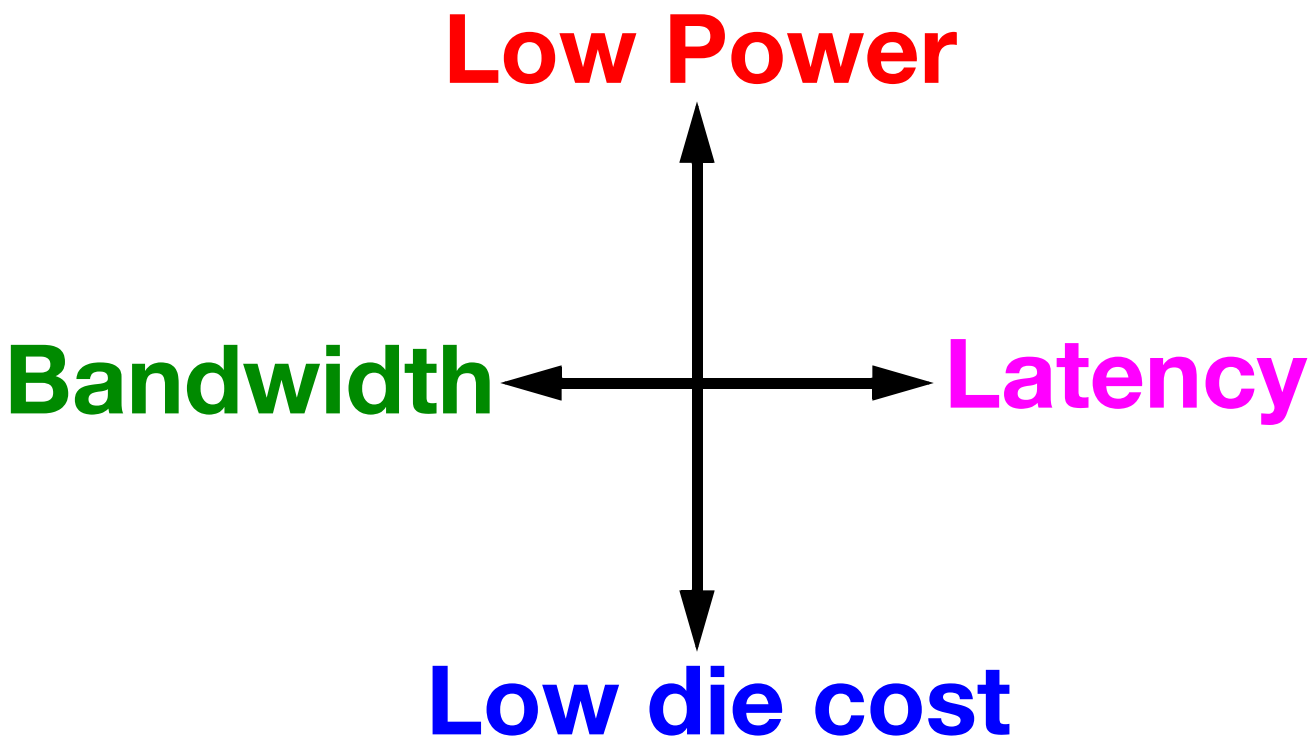
Maybe Easy is Better?



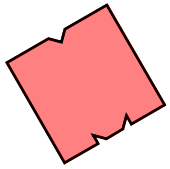
Always assume worst case.
Simple design = Small controller
Poor performance ? Does it matter?



What is the goal?

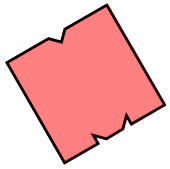


Or, little of everything?

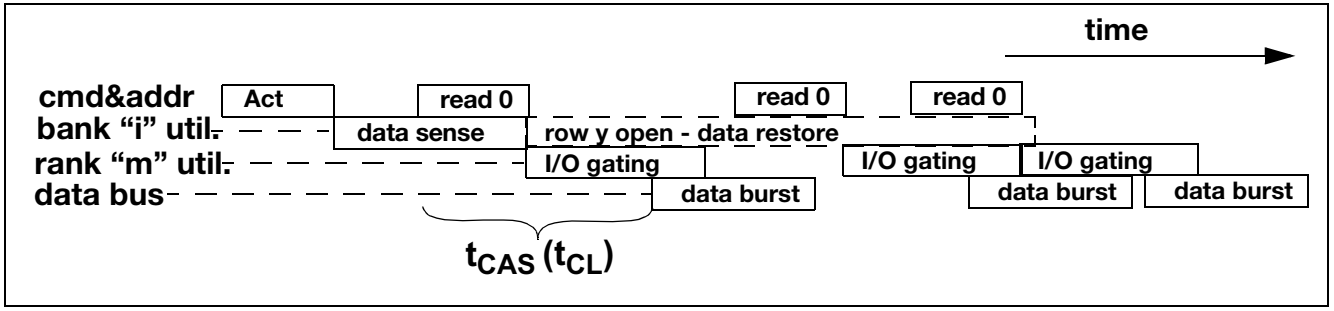


What's Important?

- **Row Buffer Management Policy**
- **Address Mapping Scheme**
- **System Configuration**
- **Workload Characteristics**
- **VA to PA translation?**
- **Command Re-ordering?**
- **Transaction Re-ordering?**



Row Buffer Mgmt: Open Page

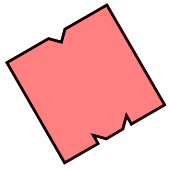


Open Page: Keep data at sense amps

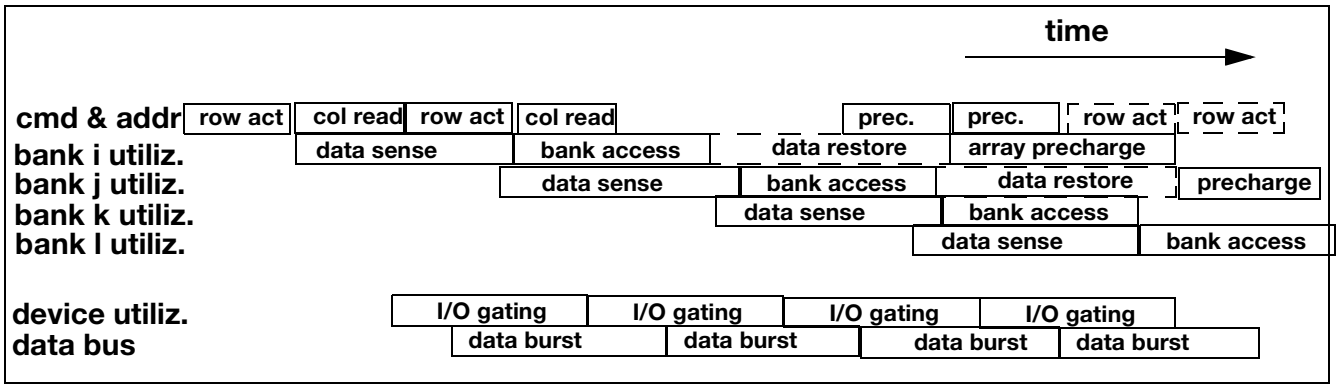
Good: If request stream has good locality
Good: If request rate is fairly high, but not too high*

Bad: Harder to schedule. More corner cases to deal with

Bad: If access rate is too low, keeping data at sense amps eat power



Row Buffer Mgmt: Close Page

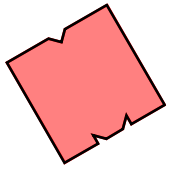


Close Page: One CAS per RAS

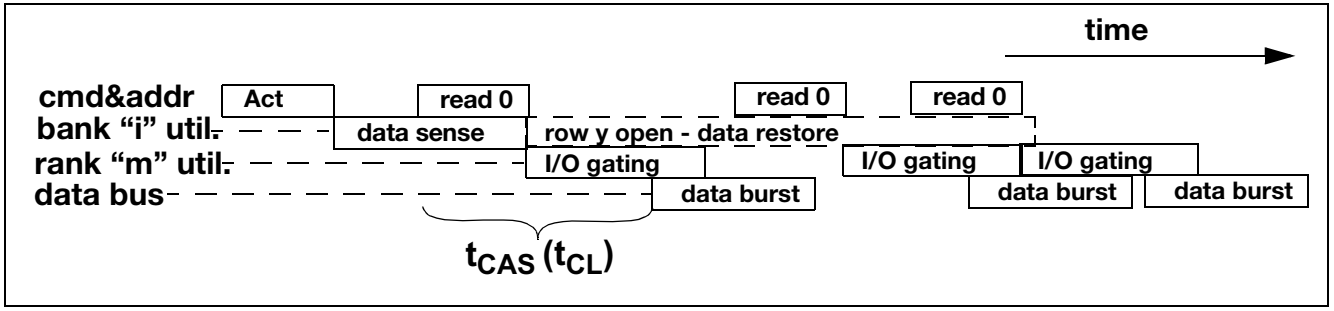
Good: If request stream has no locality
Good: If request rate is very high
Good: Easier to schedule (deep pipeline)

Bad: One CAS per RAS means high power consumption (t_{RRD} & t_{FAW})

Bad: Nightmare scenario if all accesses are to same bank



Row Buffer Mgmt: Dynamic Page

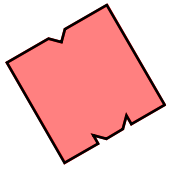


"Dynamic" Page: many CAS per RAS

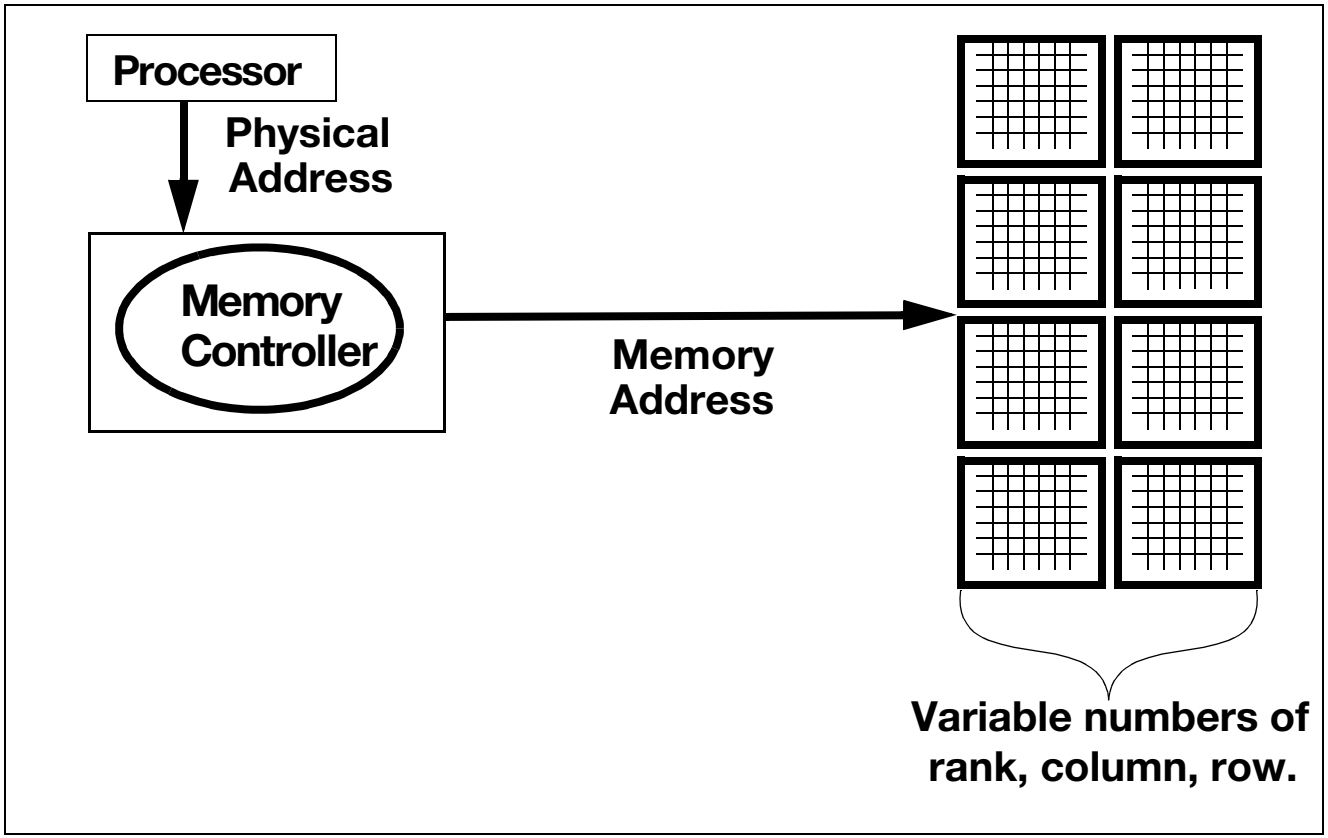
Typically based on open page mechanism

Keep timer: If timer expires, close page

Keep history: If many conflicts occur, switch to close page policy.

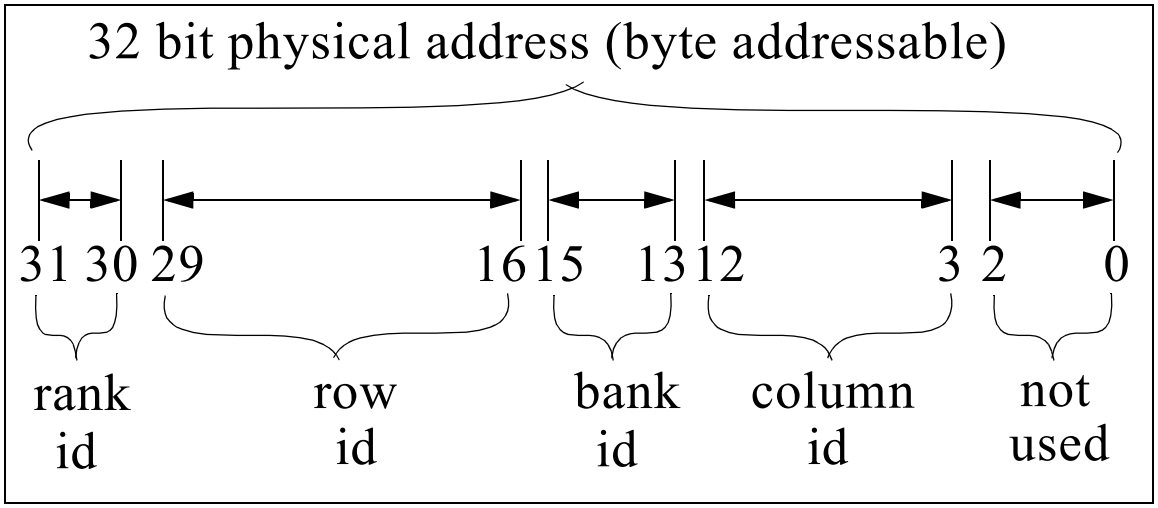


Address Mapping



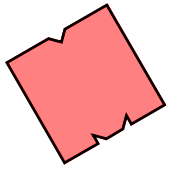
Directs requests to different locations in DRAM Memory System.

Address Mapping: Open Page

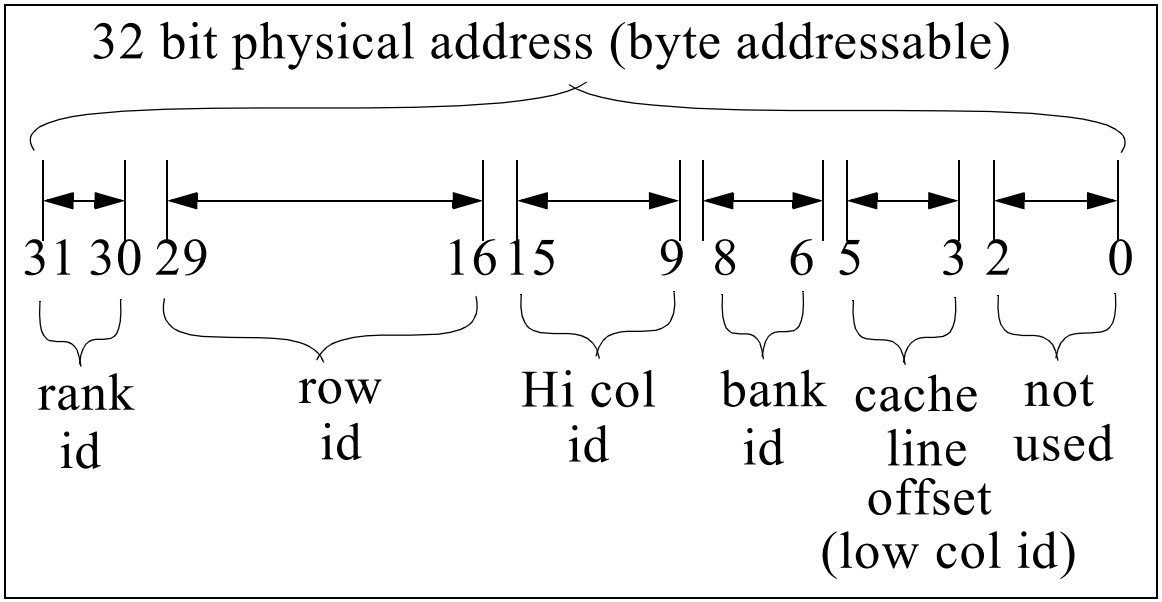


Keep cacheline i and cacheline i +1 to same bank

Using 4 ranks of 8 x8 1 Gbit DDR2 SDRAM Devices

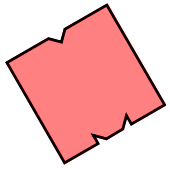


Address Mapping: Close Page

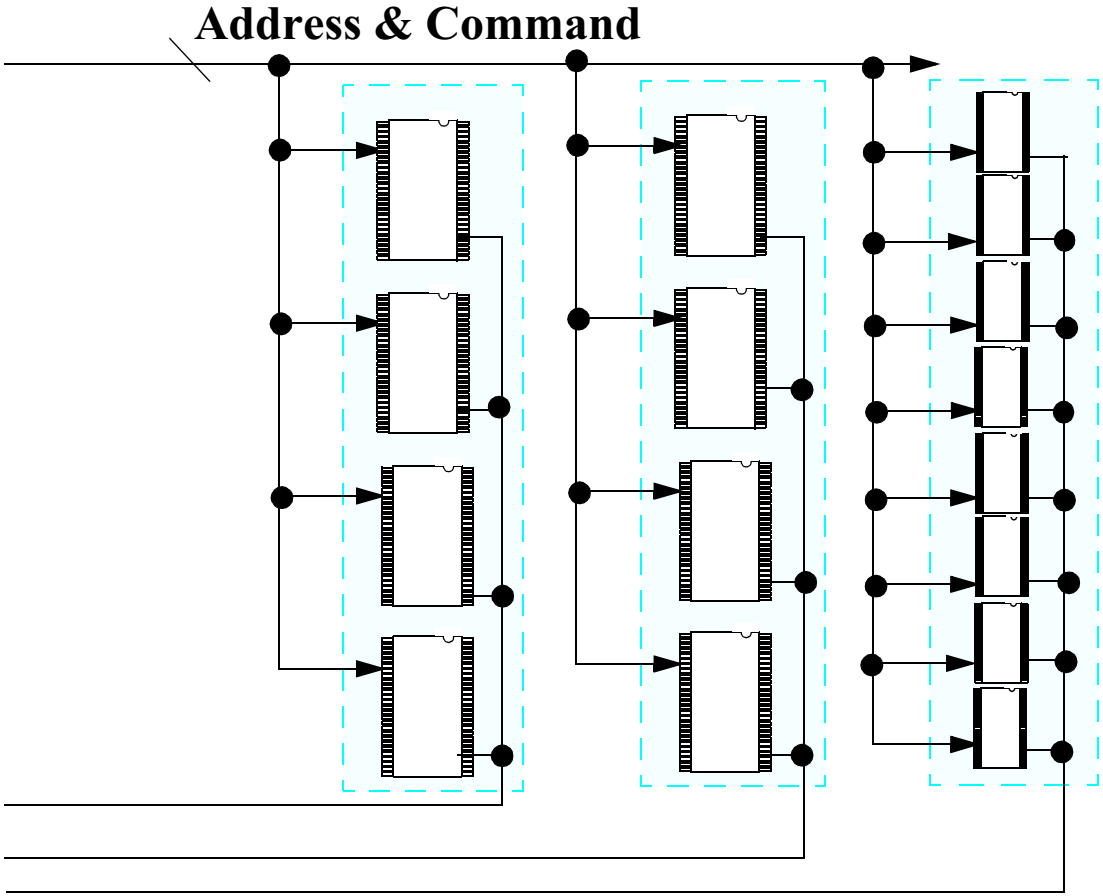


Cacheline i and cacheline i +1 are sent to different banks

Why is rank id mapped to high addr range?



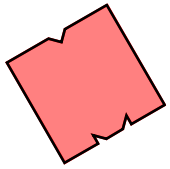
Address Mapping: Expandability I



Different organizations, density,

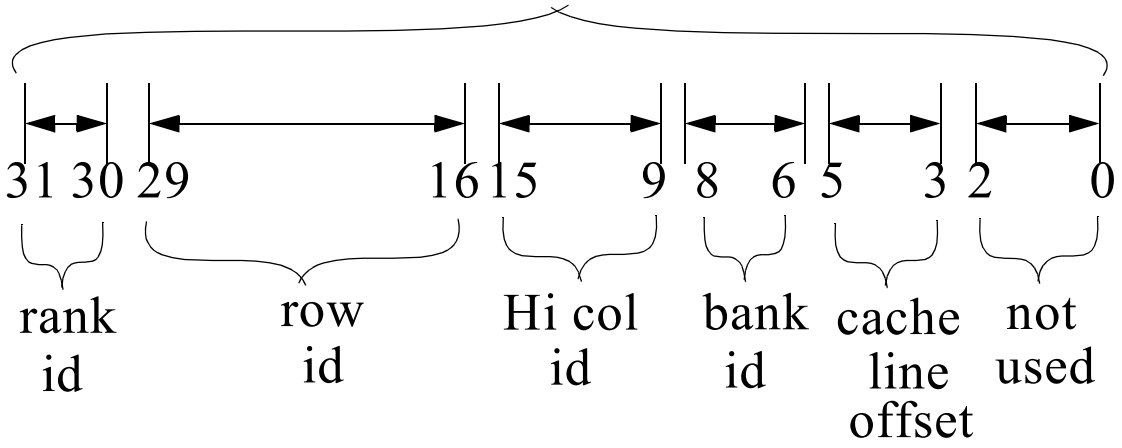
Variable number of ranks

(use address range register)

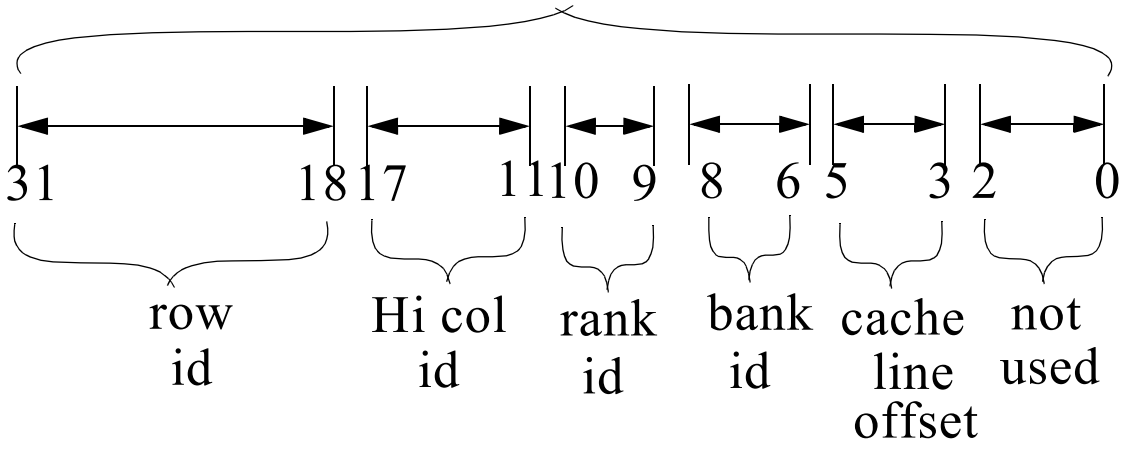


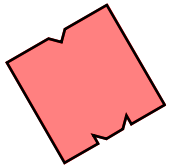
Address Mapping: Expandability II

Expandable

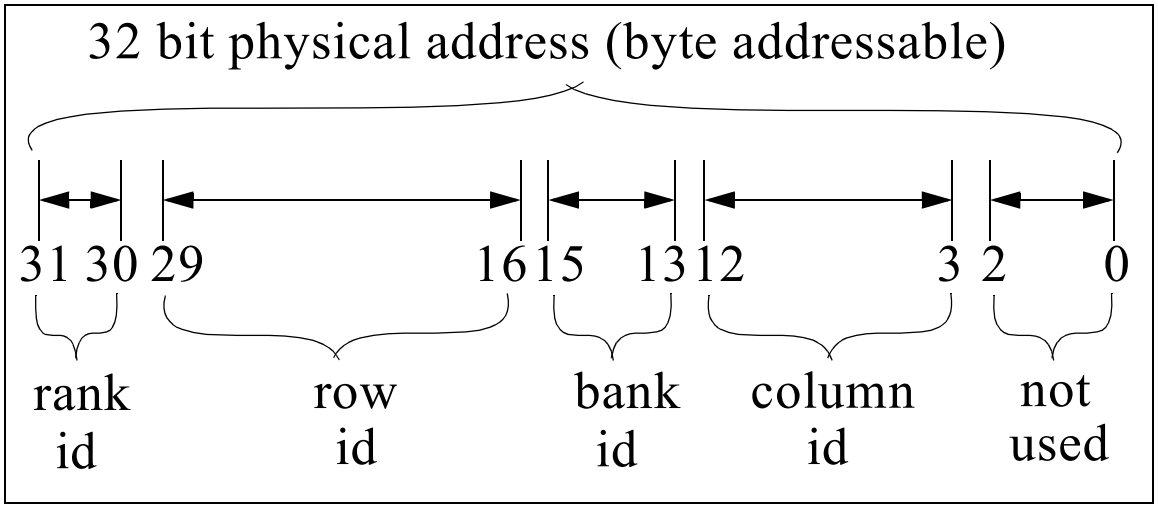


Move rank Id lower, get more parallelism, but requires matching ranks (Intel 875P - "Dynamic Addressing")





Bank Address Aliasing



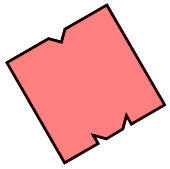
C[i] = A[i] + B[i]

Suppose that all arrays are size 2^k; K > 16

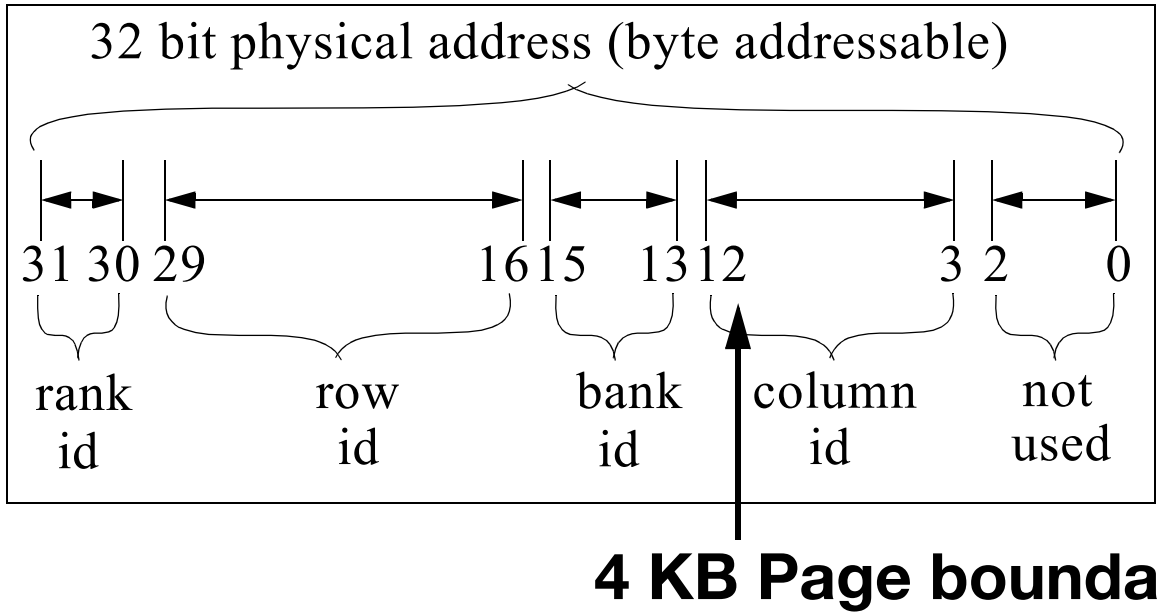
load C[0x0E1C0000];
load A[0x0E3C0000];
load B[0x0E5C0000];

Bank Conflict

Or is there?



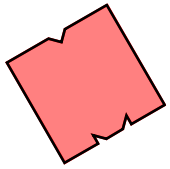
TLB VA to PA Translation



If VA to PA translation is random

load C[0x0F1E2000];
load A[0x1428B000];
load B[0x2AC23000];

Random chance for bank conflict

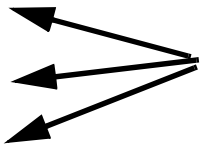


Software Solution (Offset insertion)

double A[SIZE + OFFSET];
double B[SIZE + OFFSET];
double C[SIZE + OFFSET];

All array elements offset by OFFSET

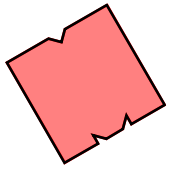
load C[0x0E1C0000];
load A[0x0E3C2000];
load B[0x0E5C4000];



No bank conflict as array marches

	Copy	Scale	Add	Triad
No Offset (MB/s)	2331	2473	2584	2496
With Offset (MB/s)	2448	2474	3164	3157
Difference (MB/s)	117	1	580	661

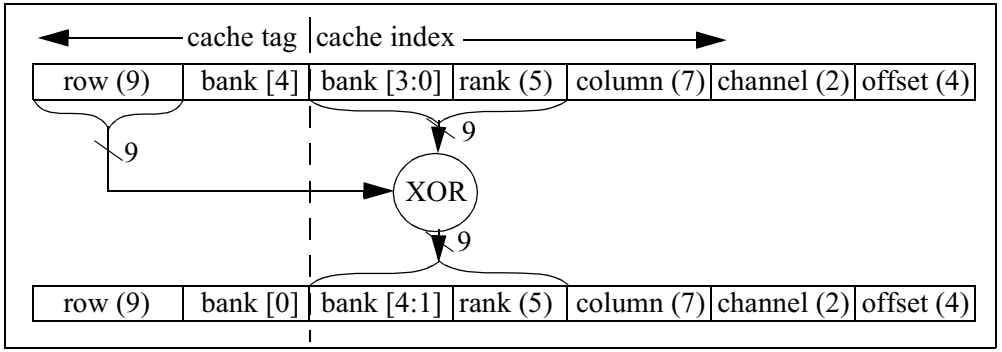
Measured on Dell Power Edge 400SC



Hardware Solution

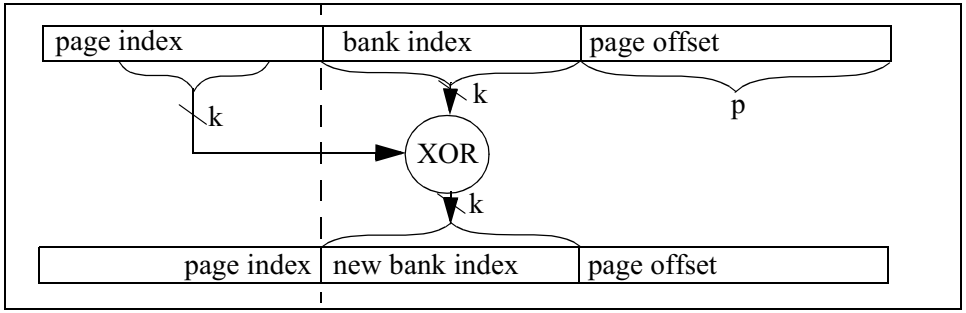
Permute physical addr before mapping to memory addr Retain 1:1 mapping characteristic, reduce bank conflicts during array marches

Rambus

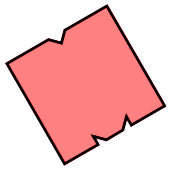


W. Lin, S. Reinhardt, D. Burger, "Reducing DRAM Latencies with an Integrated Memory Hierarchy Design". 7th International Symposium on High-Performance Computer Architecture, January 2001.

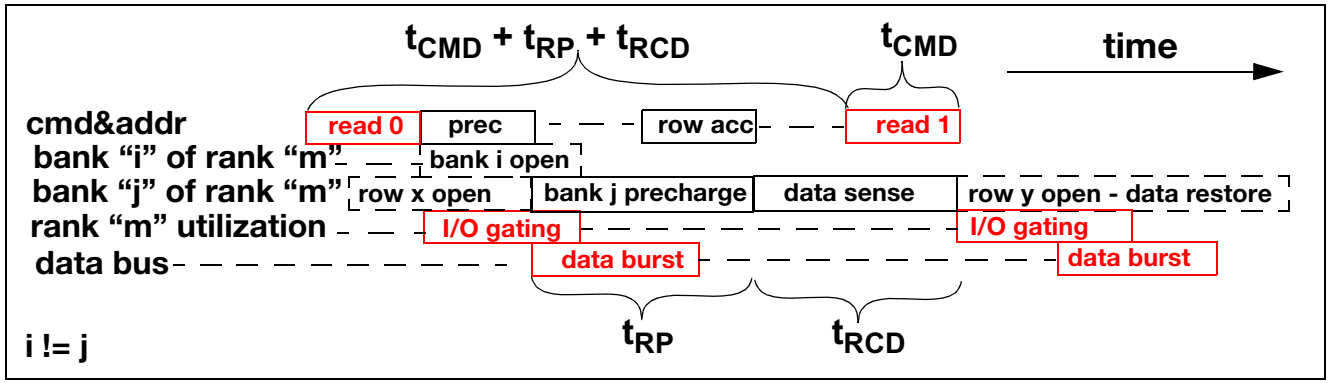
SDRAM



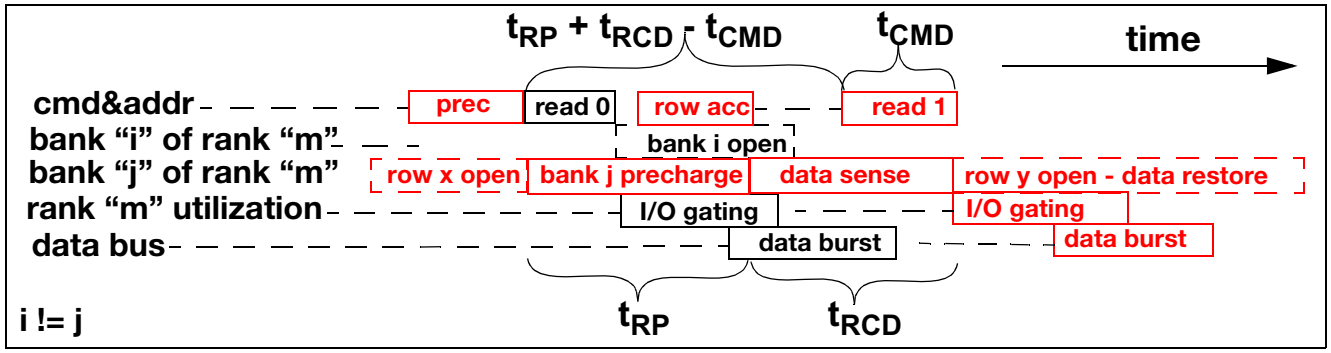
Z. Zhang, Z. Zhu, X. Zhang, "Breaking Address Mapping Symmetry at Multi-levels of Memory Hierarchy to Reduce DRAM Row-buffer Conflicts", The Journal of Instruction-Level Parallelism, Vol. 3, 2002.
Z. Zhang, Z. Zhu, X. Zhang, "A permutation-based page interleaving scheme to reduce row-buffer conflicts and exploit data locality", Proceedings of the 33rd IEEE/ACM International Symposium on Microarchitecture, Dec. 2000. pp32-41.



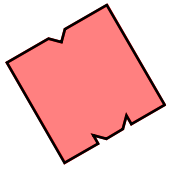
Command Re-ordering



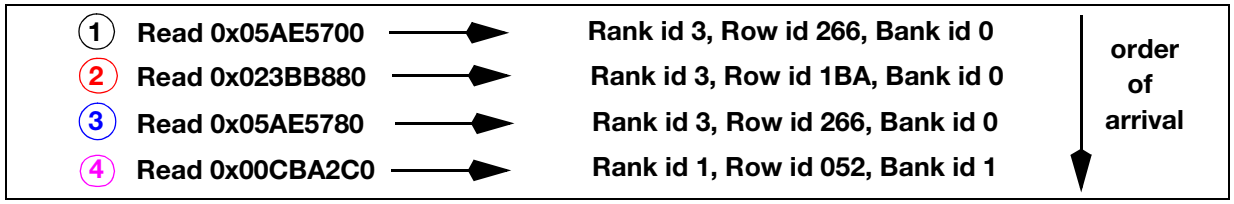
No Re-ordering : Easy to do

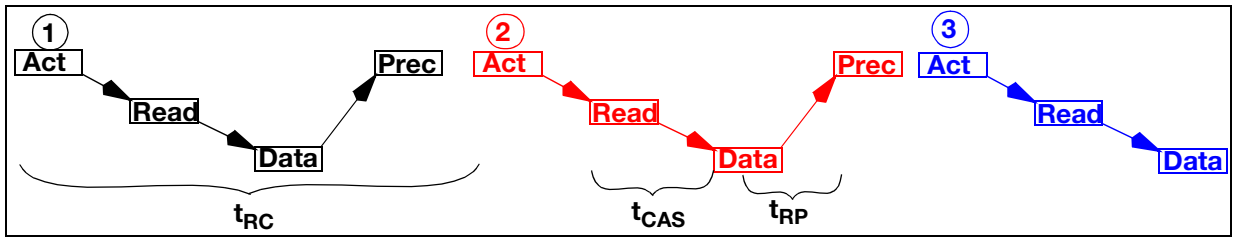


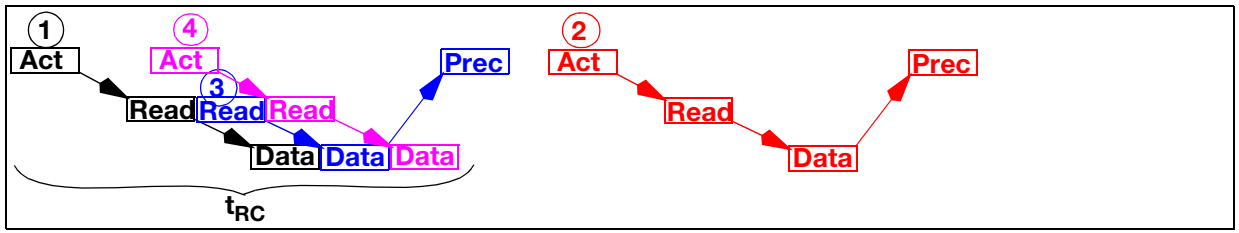
With Re-ordering: Gets harder, look to schedule opportunistically



Transaction Re-ordering



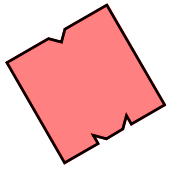




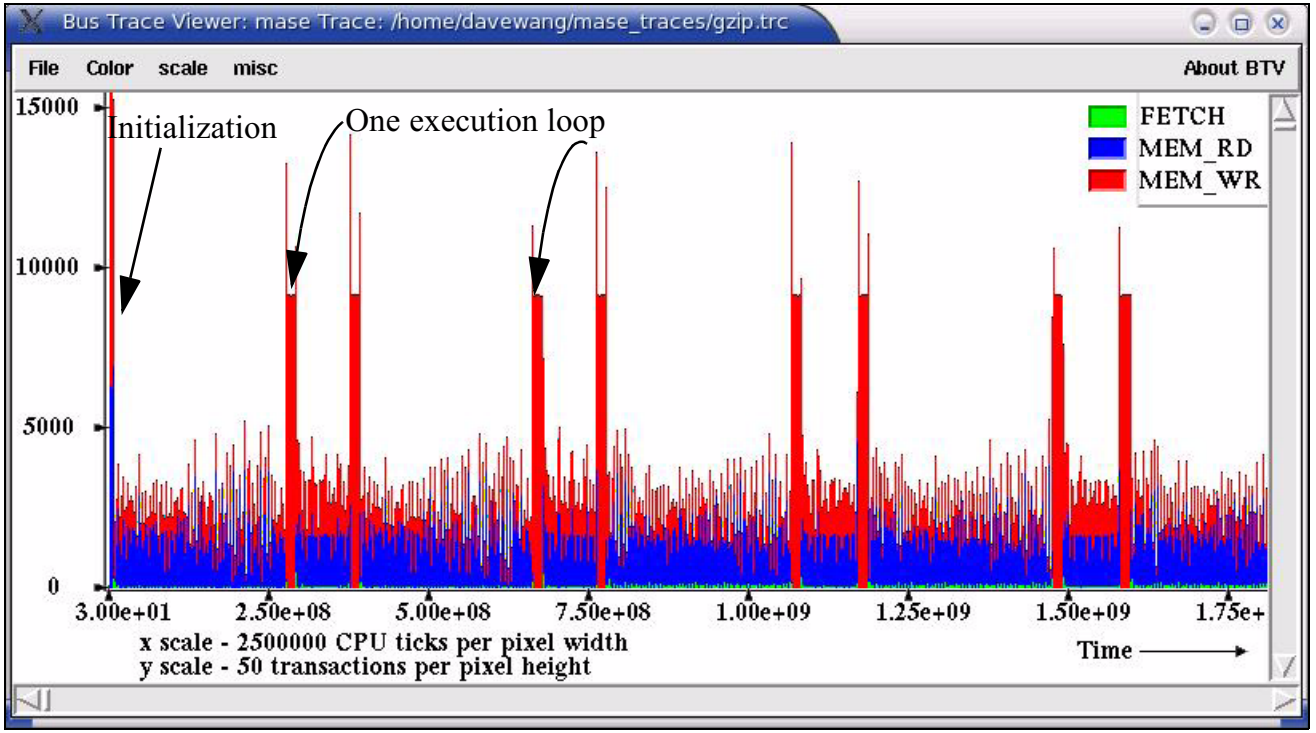
How many requests are in flight? (memory level parallelism)

How deep are the queues?

Access rate? Pattern? Locality? Type?

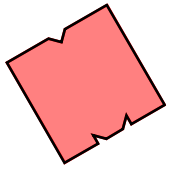


Bus Trace Viewer (BTV)

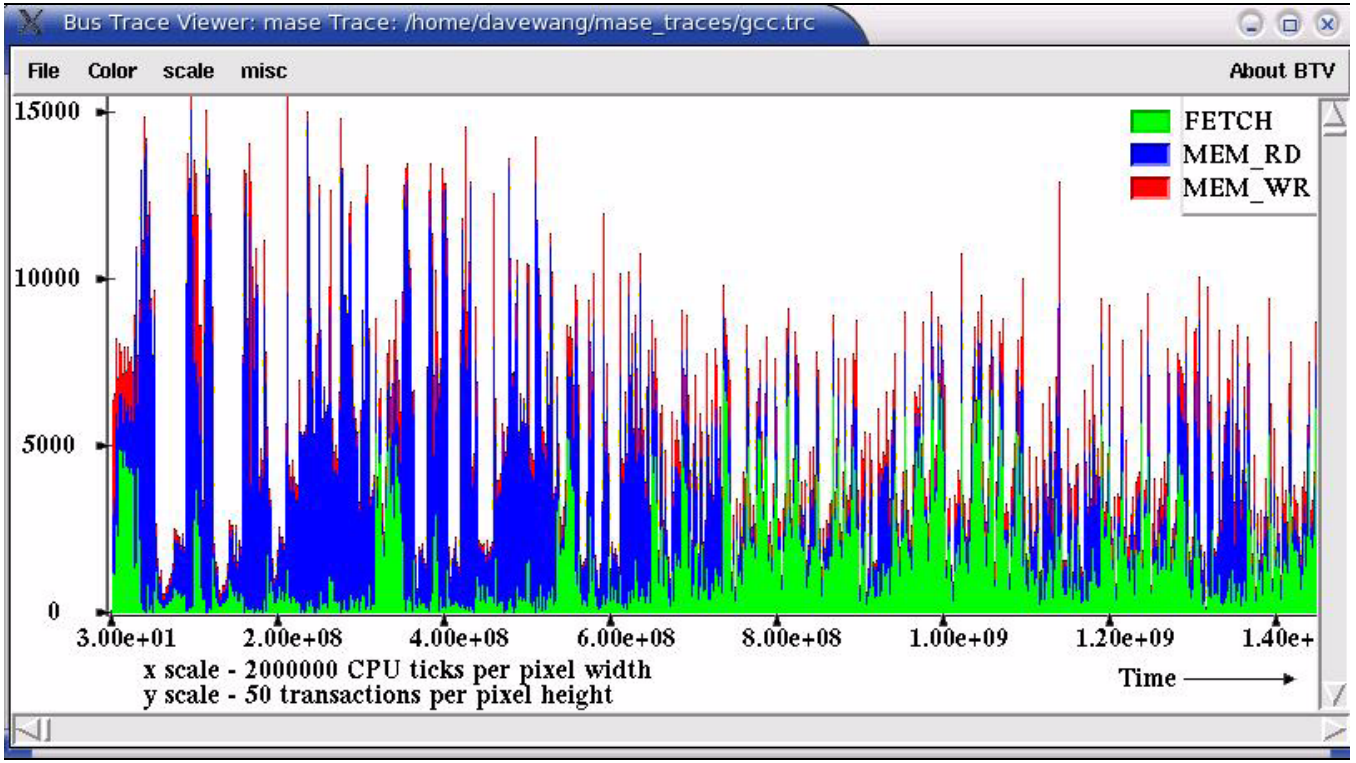


164.gzip

Written by yours truly to look at traces.

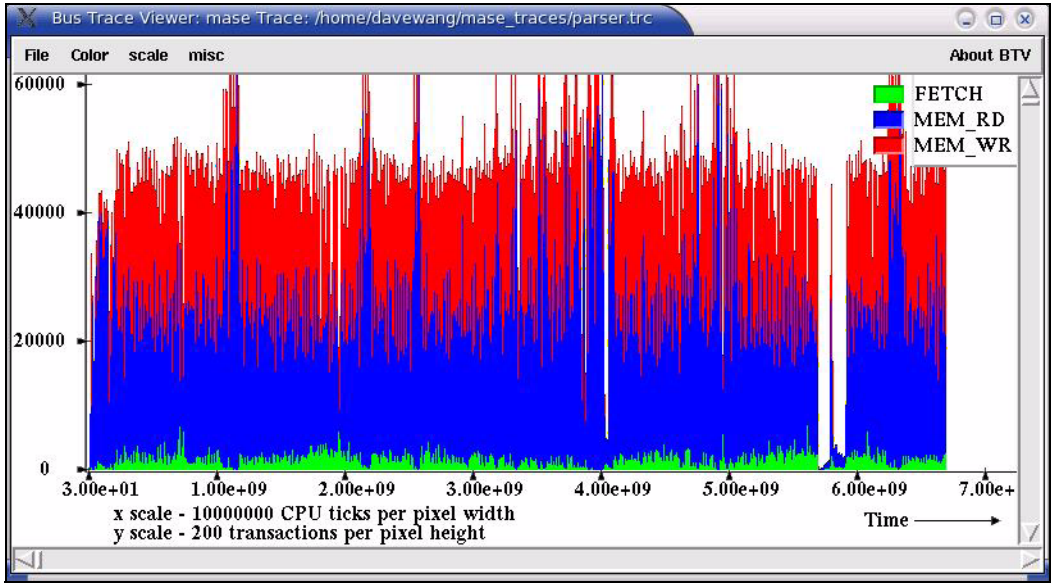


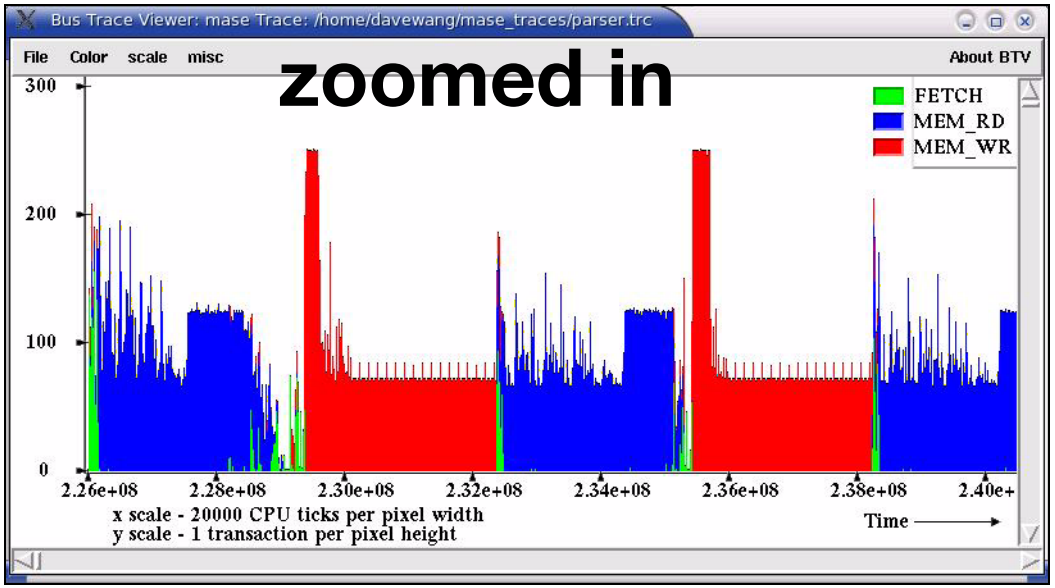
176.gcc

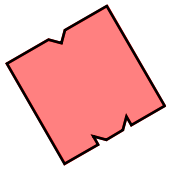


bursty, lots of IFetches (L2$ = 256KB)

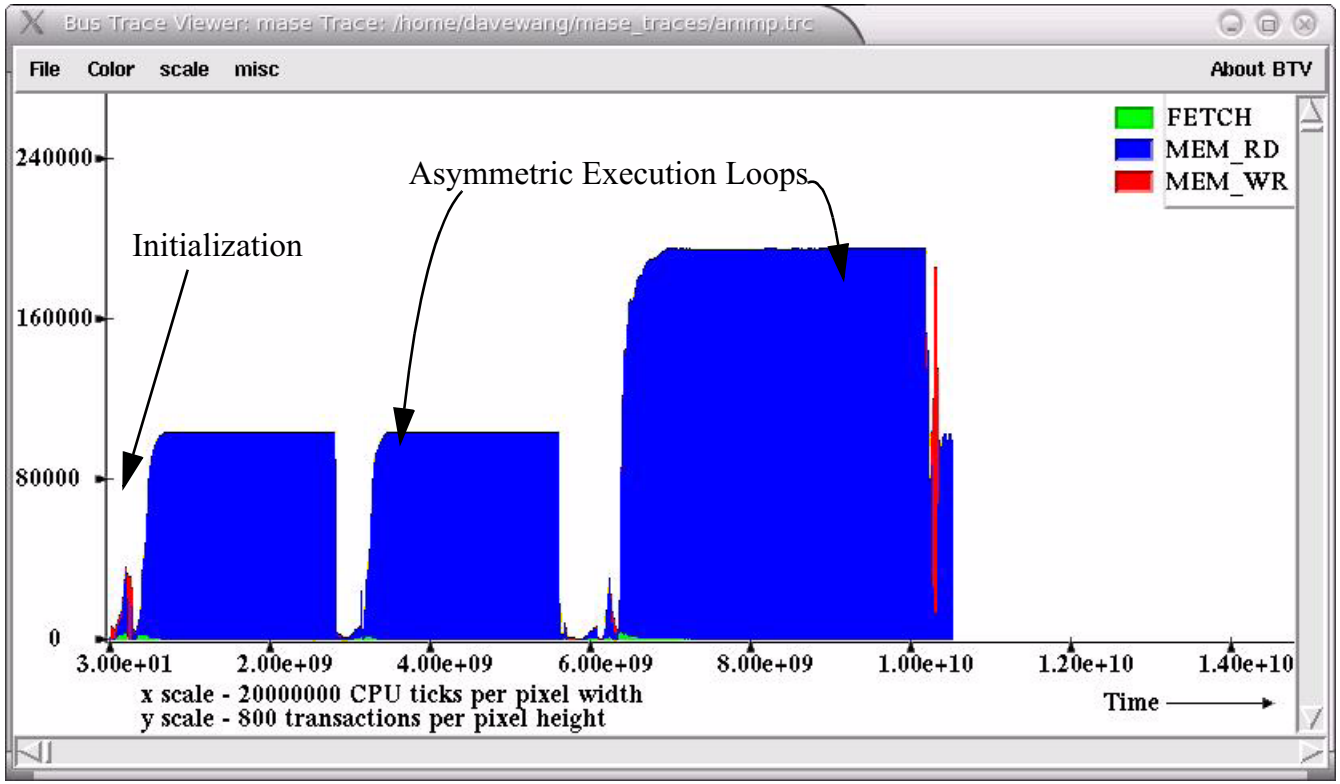
197.parser


Bus Trace Viewer: mase Trace: /home/davewang/mase_traces/parser.trc
File Color scale misc
About BTV
60000
40000
20000
0
FETCH
MEM_RD
MEM_WR
3.00e+01
1.00e+09
2.00e+09
3.00e+09
4.00e+09
5.00e+09
6.00e+09
7.00e+
x scale - 10000000 CPU ticks per pixel width
y scale - 200 transactions per pixel height
Time

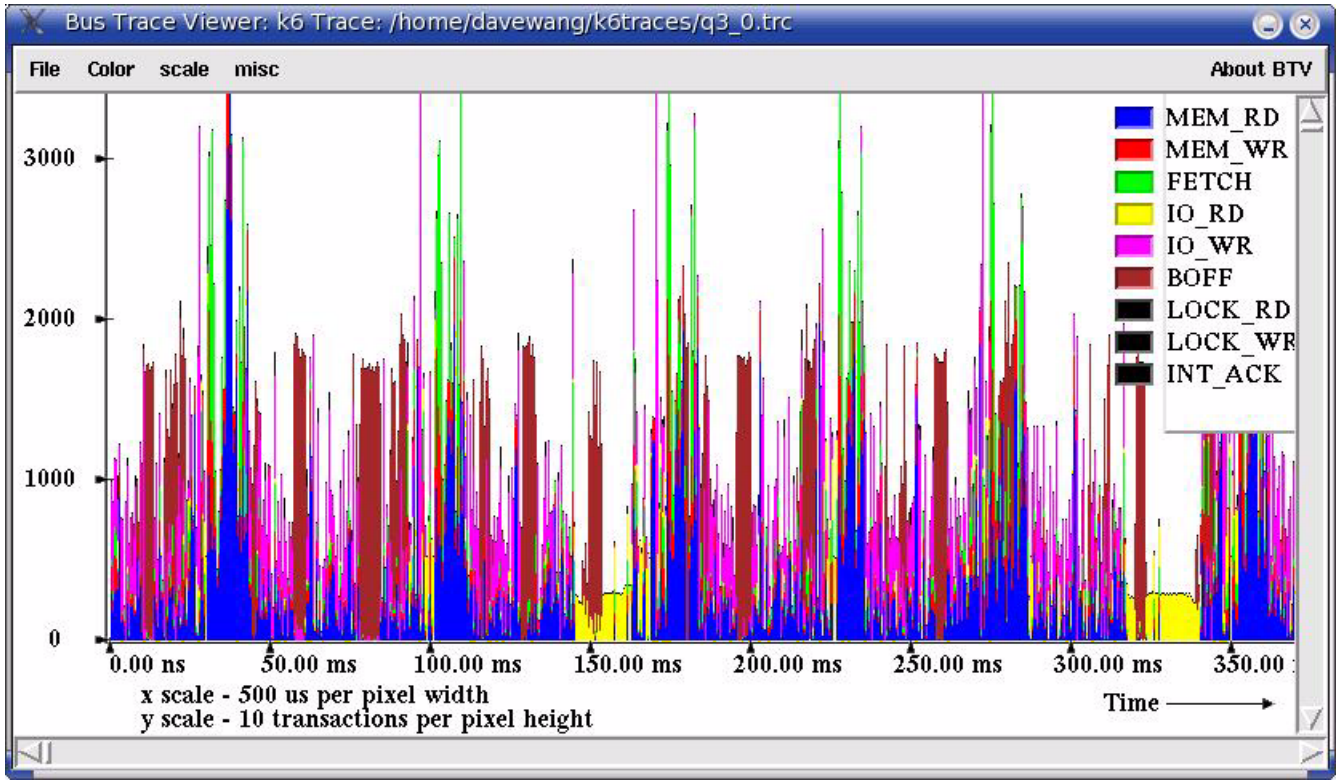

Bus Trace Viewer: mase Trace: /home/davewang/mase_traces/parser.trc
File Color scale misc
About BTV
zoomed in
300
200
100
0
FETCH
MEM_RD
MEM_WR
2.26e+08
2.28e+08
2.30e+08
2.32e+08
2.34e+08
2.36e+08
2.38e+08
2.40e+
x scale - 20000 CPU ticks per pixel width
y scale - 1 transaction per pixel height
Time



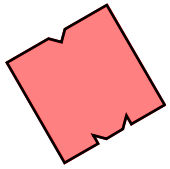
188.ammp


Bus Trace Viewer: mase Trace: /home/davewang/mase_traces/ammp.trc
File
Color
scale
misc
About BTV
FETCH
MEM_RD
MEM_WR
240000
160000
80000
0
Initialization
Asymmetric Execution Loops
3.00e+01
2.00e+09
4.00e+09
6.00e+09
8.00e+09
1.00e+10
1.20e+10
1.40e+10
x scale - 20000000 CPU ticks per pixel width
y scale - 800 transactions per pixel height
Time

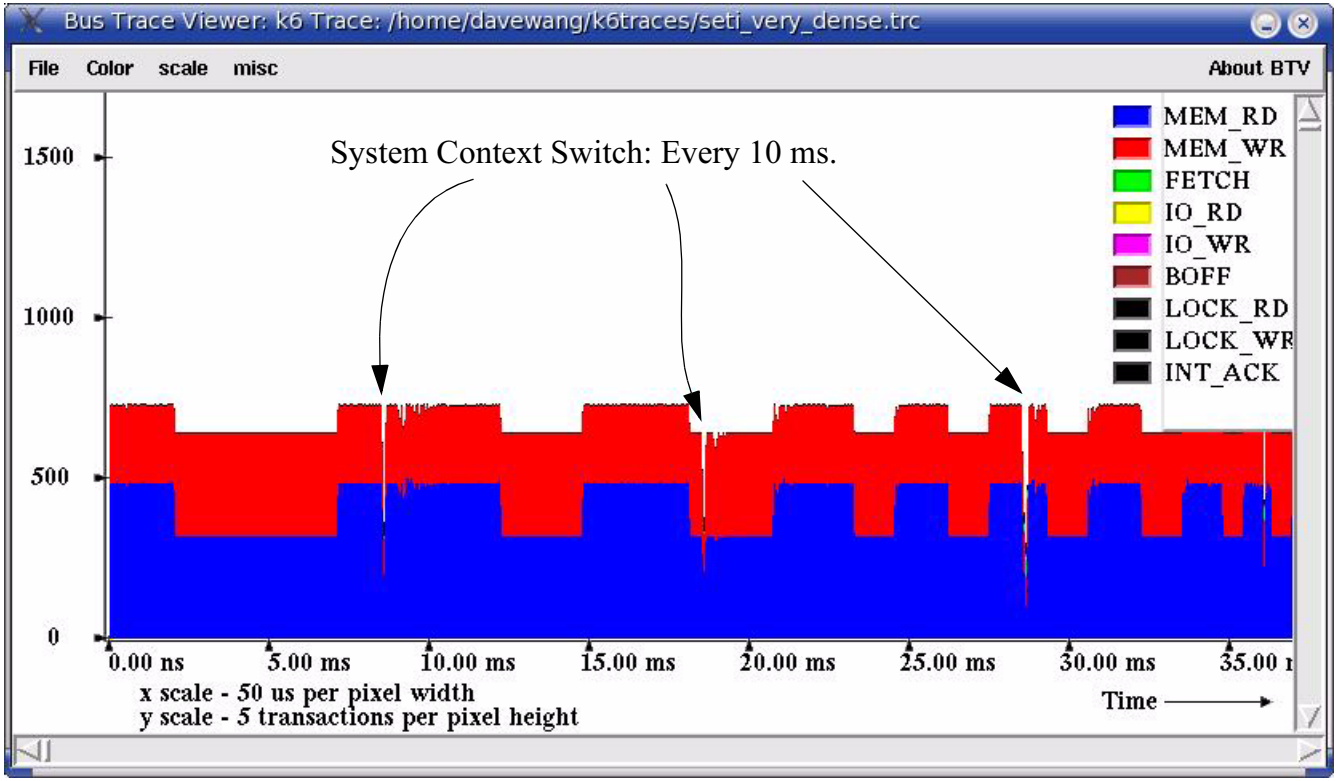
Quake 3 (random segment 0)

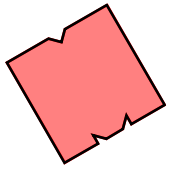


one frame

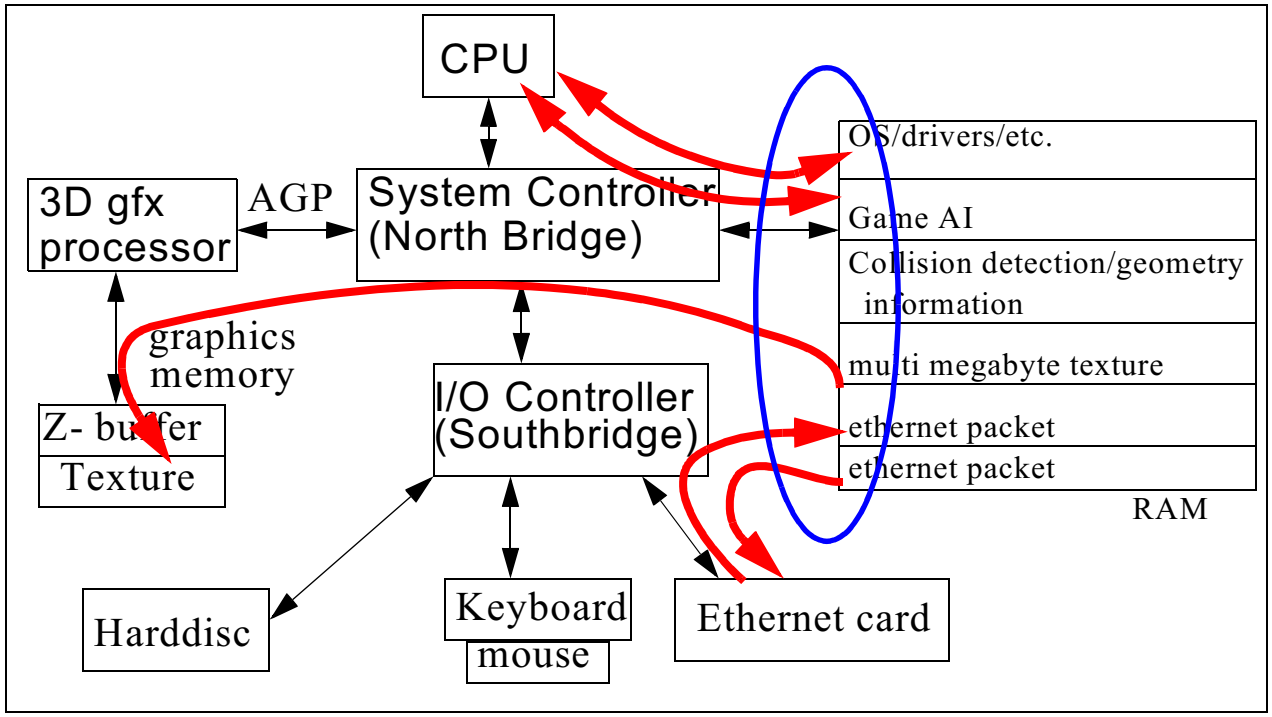


SETI@HOME


Bus Trace Viewer: k6 Trace: /home/davewang/k6traces/seti_very_dense.trc
File
Color
scale
misc
About BTV
System Context Switch: Every 10 ms.
MEM_RD
MEM_WR
FETCH
IO_RD
IO_WR
BOFF
LOCK_RD
LOCK_WR
INT_ACK
1500
1000
500
0
0.00 ns
5.00 ms
10.00 ms
15.00 ms
20.00 ms
25.00 ms
30.00 ms
x scale - 50 us per pixel width
y scale - 5 transactions per pixel height
Time



Recall


CPU
3D gfx processor
AGP
System Controller (North Bridge)
OS/drivers/etc.
Game AI
Collision detection/geometry information
multi megabyte texture
ethernet packet
ethernet packet
RAM
graphics memory
Z- buffer
Texture
I/O Controller (Southbridge)
Harddisc
Keyboard
mouse
Ethernet card

What about MP/CMP/MT?