

Mainstream Computer System Components

Double Date Rate (DDR) SDRAM
One channel = 8 bytes = 64 bits wide

Current DDR3 SDRAM Example:

PC3-12800 (DDR3-1600) ←
200 MHz (internal base chip clock)
8-way interleaved (8-banks)
~12.8 GBYTES/SEC (peak)
(one 64bit channel)
~25.6 GBYTES/SEC (peak)
(two 64bit channels – e.g AMD x4, x6)
~38.4 GBYTES/SEC (peak)
(three 64bit channels – e.g Intel Core i7)

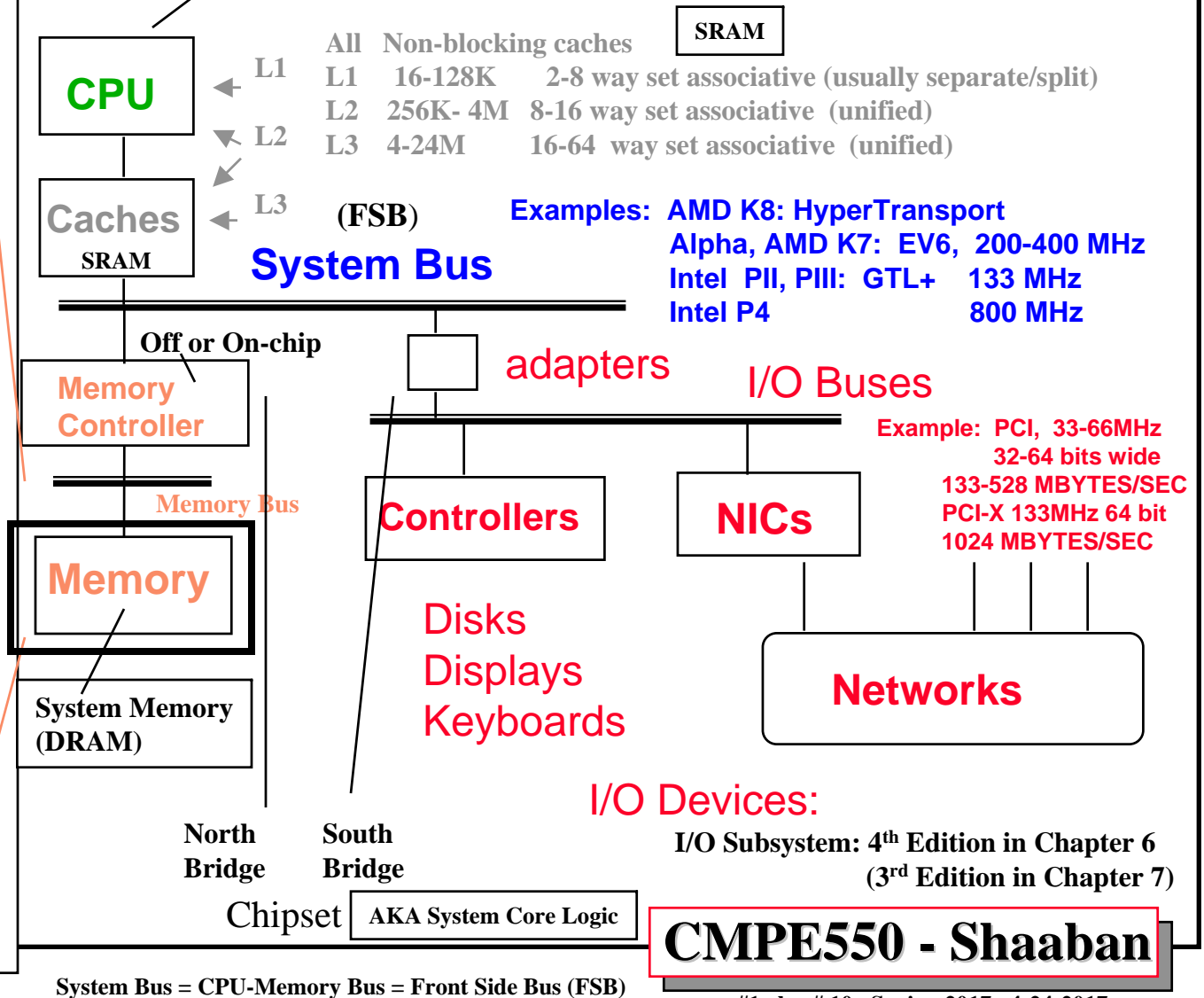
PC2-6400 (DDR2-800)
200 MHz (internal base chip clock)
64-128 bits wide
4-way interleaved (4-banks)
~6.4 GBYTES/SEC (peak)
(one 64bit channel)
~12.8 GBYTES/SEC (peak)
(two 64bit channels)

DDR SDRAM Example:
PC3200 (DDR-400)
200 MHz (base chip clock)
4-way interleaved (4-banks)
~3.2 GBYTES/SEC (peak)
(one 64bit channel)
~6.4 GBYTES/SEC
(two 64bit channels)

Single Date Rate SDRAM
PC100/PC133
100-133MHz (base chip clock)
64-128 bits wide
2-way intelevaed (2-banks)
~ 900 MBYTES/SEC peak (64bit)

CPU Core 2 GHz - 3.5 GHz 4-way Superscaler (RISC or RISC-core (x86):
Dynamic scheduling, Hardware speculation
Multiple FP, integer FUs, Dynamic branch prediction ...

One core or multi-core (2-8) per chip



System Bus = CPU-Memory Bus = Front Side Bus (FSB)


CMPE550 - Shaaban

The Memory Hierarchy

- **Review of Memory Hierarchy & Cache Basics (from 350)**
 - Cache Basics:
 - CPU Performance Evaluation with Cache
- **Classification of Steady-State Cache Misses:**
 - *The Three C's of cache Misses*
- **Cache Write Policies/Performance Evaluation**
- **Cache Write Miss Policies**
- **Multi-Level Caches & Performance**

Cache exploits access locality to:

- 1 Lower AMAT by hiding long main memory access latency.
- 2 Lower demands on main memory bandwidth.

- 
- **Main Memory:**
 - Performance Metrics: Latency & Bandwidth
 - Key DRAM Timing Parameters
 - DRAM System Memory Generations
 - Basic Memory Bandwidth Improvement/Miss Penalty Reduction Techniques

4th Edition: Chapter 5.3
3rd Edition: Chapter 5.8, 5.9

- **Techniques To Improve Cache Performance:**
 - Reduce Miss Rate
 - Reduce Cache Miss Penalty
 - Reduce Cache Hit Time
- **Virtual Memory**
 - Benefits, Issues/Strategies
 - Basic Virtual → Physical Address Translation: Page Tables
 - Speeding Up Address Translation: Translation Lookaside Buffer (TLB)

i.e Memory latency reduction

CMPE550 - Shaaban

Addressing The CPU/Memory Performance Gap:

Memory Access Latency Reduction & Hiding Techniques

Memory Latency Reduction Techniques:

Reduce it!

- Faster Dynamic RAM (DRAM) Cells: Depends on VLSI processing technology.

- Wider Memory Bus Width/Interface: Fewer memory bus accesses needed

- Burst Mode Memory Access

e.g 128 vs. 64 bits

- Multiple Memory Banks:

- At DRAM chip level (SDR, DDR, DDR2, DDR3 SDRAM), module or channel levels.

Basic Memory Bandwidth
Improvement/Miss Penalty
Reduction Techniques

- Integration of Memory Controller with Processor: e.g AMD and Intel's current processor architecture

- New Emerging "Faster" RAM Technologies:

- New Types of RAM Cells: e.g. Magnetoresistive RAM (MRAM), Zero-capacitor RAM (Z-RAM), Thyristor RAM (T-RAM) ...
- 3D-Stacked Memory: e.g Micron's Hybrid Memory Cube (HMC), AMD's High Bandwidth Memory (HBM).

Memory Latency Hiding Techniques:

Hide it!

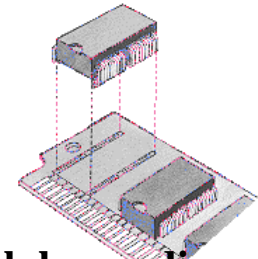
Lecture 8

- Memory Hierarchy: One or more levels of smaller and faster memory (SRAM-based cache) on- or off-chip that exploit program access locality to hide long main memory latency.

- Pre-Fetching: Request instructions and/or data from memory before actually needed to hide long memory access latency.

CMPE550 - Shaaban

Main Memory



- Main (or system) memory generally utilizes Dynamic RAM (DRAM), which use a single transistor to store a bit, but require a periodic data refresh by reading every row increasing cycle time. DRAM: Slow but high density
- Static RAM may be used for main memory if the added expense, low density, high power consumption, and complexity is feasible (e.g. Cray Vector Supercomputers). SRAM: Fast but low density
- Main memory performance is affected by:
 - **Memory latency:** Affects cache miss penalty, M. Measured by:
 - **Memory Access time:** The time it takes between a memory access request is issued to main memory and the time the requested information is available to cache/CPU.
 - **Memory Cycle time:** The minimum time between requests to memory (greater than access time in DRAM to allow address lines to be stable)
 - **Peak or Nominal Memory bandwidth:** The maximum sustained data transfer rate between main memory and cache/CPU.
 - In current memory technologies (e.g Double Data Rate SDRAM) published peak memory bandwidth does not take account most of the memory access latency.
 - This leads to achievable realistic memory bandwidth < peak memory bandwidth

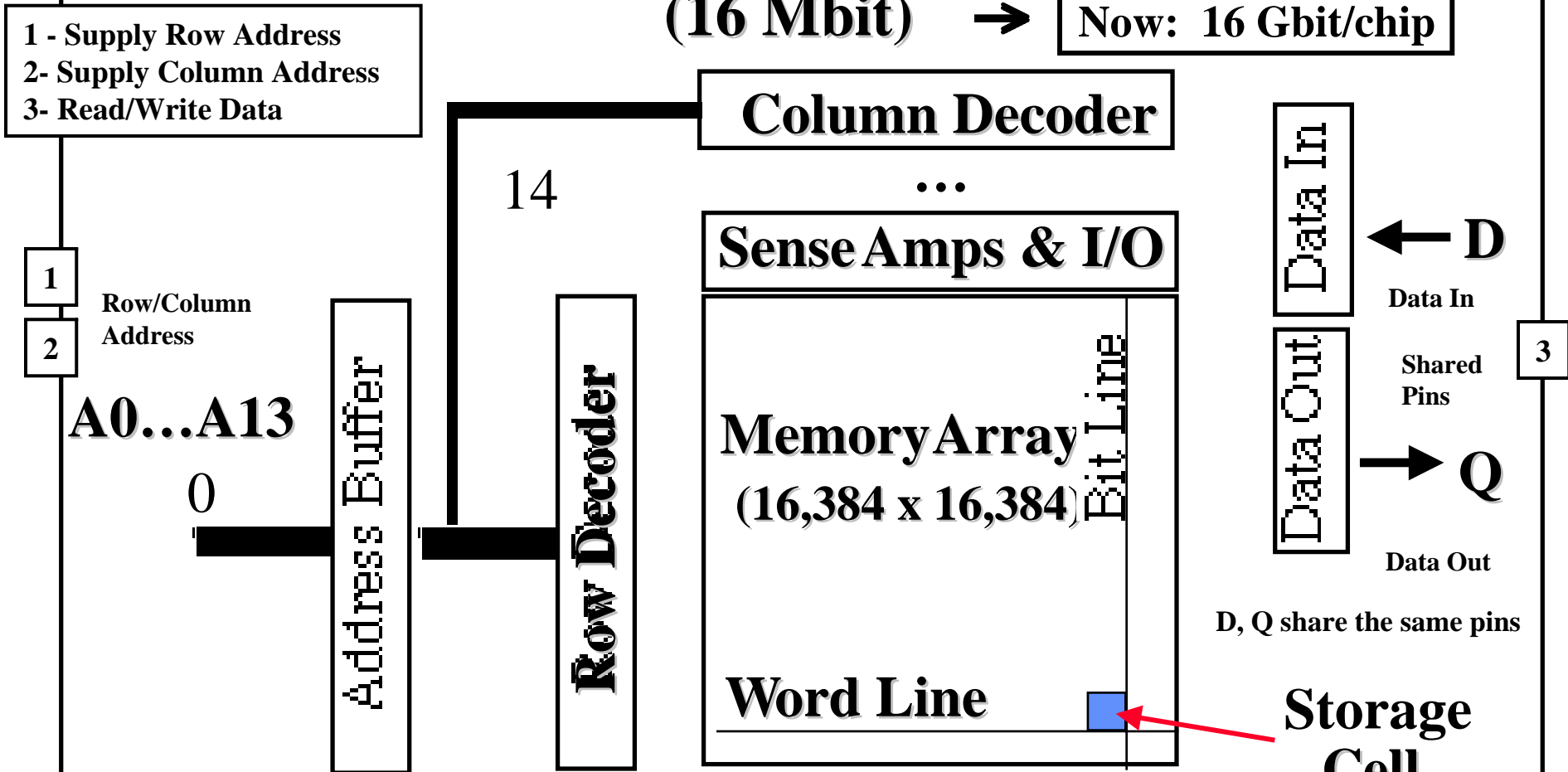
4th Edition: Chapter 5.3
3rd Edition: Chapter 5.8, 5.9

Or maximum effective memory bandwidth

CMPE550 - Shaaban

Logical Dynamic RAM (DRAM) Chip Organization

(16 Mbit) → Now: 16 Gbit/chip



Basic Steps:

Control Signals:

- 1 - Row Access Strobe (RAS): Low to latch row address
- 2- Column Address Strobe (CAS): Low to latch column address
- 3- Write Enable (WE) or Output Enable (OE)
- 4- Wait for data to be ready

A periodic data refresh is required by reading every bit

1 - Supply Row Address 2- Supply Column Address 3- Get Data

CMPE550 - Shaaban

Four Key DRAM Timing Parameters

- 1 • **t_{RAC}** : Minimum time from RAS (Row Access Strobe) line falling (activated) to the valid data output.
 - Used to be quoted as the nominal speed of a DRAM chip
 - For a typical 64Mb DRAM $t_{\text{RAC}} = 60$ ns
- 2 • **t_{RC}** : Minimum time from the start of one row access to the start of the next (memory cycle time).
 - $t_{\text{RC}} = t_{\text{RAC}} + \text{RAS Precharge Time}$
 - $t_{\text{RC}} = 110$ ns for a 64Mbit DRAM with a t_{RAC} of 60 ns
- 3 • **t_{CAC}** : Minimum time from CAS (Column Access Strobe) line falling to valid data output.
 - 12 ns for a 64Mbit DRAM with a t_{RAC} of 60 ns
- 4 • **t_{PC}** : Minimum time from the start of one column access to the start of the next.
 - $t_{\text{PC}} = t_{\text{CAC}} + \text{CAS Precharge Time}$
 - About 25 ns for a 64Mbit DRAM with a t_{RAC} of 60 ns

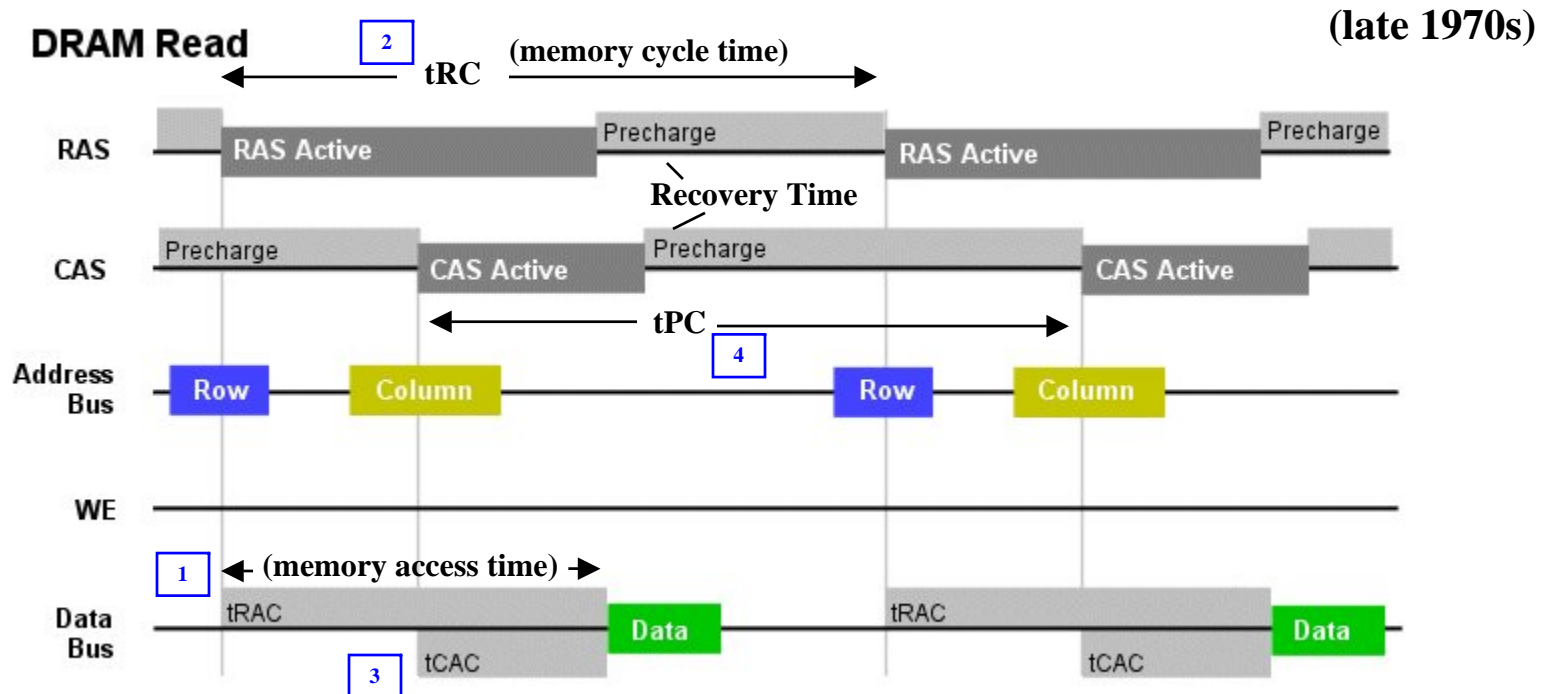
1 - Supply Row Address 2- Supply Column Address 3- Get Data

CMPE550 - Shaaban

Simplified Asynchronous DRAM Read Timing

$$\text{Memory Cycle Time} = t_{RC} = t_{RAC} + \text{RAS Precharge Time}$$

Non-burst Mode Memory Access Example



- 1 t_{RAC} : Minimum time from RAS (Row Access Strobe) line falling to the valid data output.
- 2 t_{RC} : Minimum time from the start of one row access to the start of the next (memory cycle time).
- 3 t_{CAC} : minimum time from CAS (Column Access Strobe) line falling to valid data output.
- 4 t_{PC} : minimum time from the start of one column access to the start of the next.

Peak Memory Bandwidth = Memory bus width / Memory cycle time

Example: Memory Bus Width = 8 Bytes Memory Cycle time = 200 ns

Peak Memory Bandwidth = $8 / 200 \times 10^{-9} = 40 \times 10^6$ Bytes/sec

CMPE550 - Shaaban

Simplified DRAM Speed Parameters

- **Row Access Strobe (RAS) Time:** (similar to t_{RAC}):

- Minimum time from RAS (Row Access Strobe) line falling (activated) to the first valid data output.
- A major component of memory latency. And cache miss penalty M
- Only improves ~ 5% every year.

Effective

- **Column Access Strobe (CAS) Time/data transfer time:** (similar to t_{CAC})

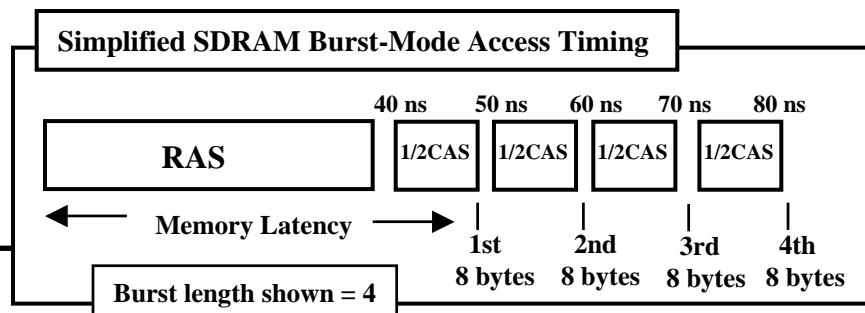
- The minimum time required to read additional data by changing column address while keeping the same row address. Burst-Mode Access
- Along with memory bus width, determines peak memory bandwidth.

- e.g For SDRAM Peak Memory Bandwidth = Bus Width / (0.5 x t_{CAC})

Example

For PC100 SDRAM Memory bus width = 8 bytes $t_{CAC} = 20\text{ns}$

Peak Bandwidth = $8 \times 100 \times 10^6 = 800 \times 10^6$ bytes/sec



For PC100 SDRAM: Clock = 100 MHz

CMPE550 - Shaaban

DRAM Generations

| Year | Chip Density (bits/chip) Size | RAS (ns) | Effective CAS (ns) | ~ RAS+ Cycle Time | Memory Type | |
|------|-------------------------------------|------------------------------------|------------------------------------|--------------------------------|-------------------|-------------------|
| 1980 | 64 Kb | 150-180 | 75 | 250 ns | Page Mode | Asynchronous DRAM |
| 1983 | 256 Kb | 120-150 | 50 | 220 ns | Page Mode | |
| 1986 | 1 Mb | 100-120 | 25 | 190 ns | | |
| 1989 | 4 Mb | 80-100 | 20 | 165 ns | Fast Page Mode | |
| 1992 | 16 Mb | 60-80 | 15 | 120 ns | EDO | |
| 1996 | 64 Mb | 50-70 | 12 | 110 ns | PC66 SDRAM | Synchronous DRAM |
| 1998 | 128 Mb | 50-70 | 10 | 100 ns | PC100 SDRAM | |
| 2000 | 256 Mb | 45-65 | 7 | 90 ns | PC133 SDRAM | |
| 2002 | 512 Mb | 40-60 | 5 | 80 ns | PC2700 DDR SDRAM | |
| | | | | | | |
| | ↓ | 8000:1 (Capacity) | 15:1 (~bandwidth) | 3:1 (Latency) | ↓ | |
| 2013 | 8 Gb | | Peak | | PC3200 DDR (2003) | |
| 2016 | 16 Gb | | | | ↓ | |
| | | | | | DDR2 SDRAM (2004) | |
| | | | | | ↓ | |
| | | | | | DDR3 SDRAM (2007) | |
| | | | | | ↓ | |
| | | | | | DDR4 SDRAM (2014) | |

A major factor in cache miss penalty M

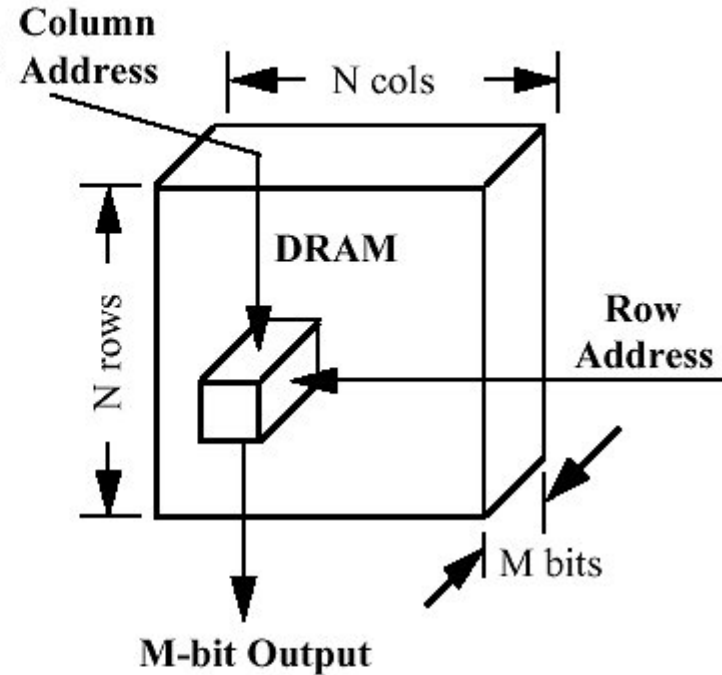
Asynchronous DRAM:

Page Mode DRAM (Early 1980s)

Last system memory type to use non-burst access mode

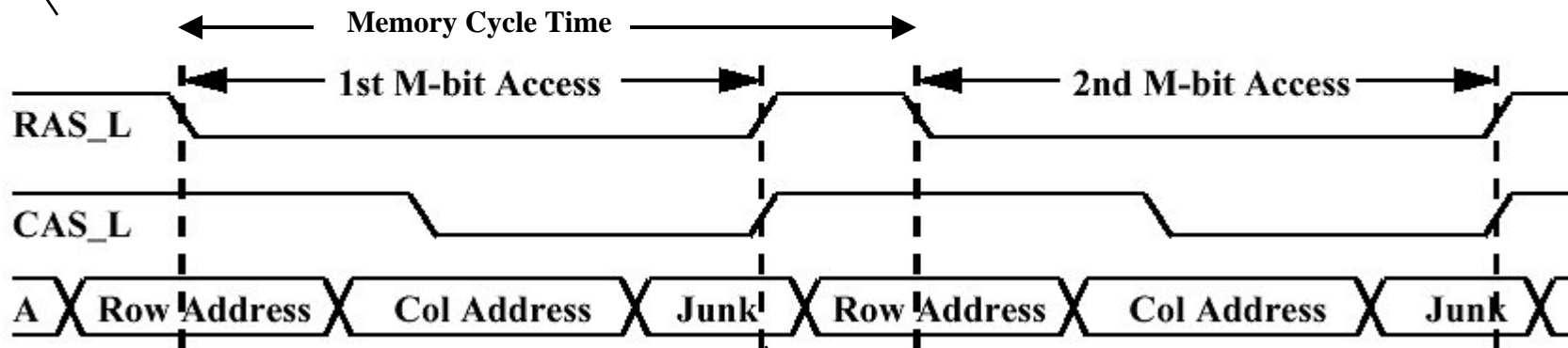
Regular DRAM Organization:

- N rows x N column x M-bit
- Read & Write M-bit at a time
- Each M-bit access requires a RAS / CAS cycle



- 1 - Supply Row Address
- 2 - Supply Column Address
- 3 - Read/Write Data

Non-burst Mode Memory Access



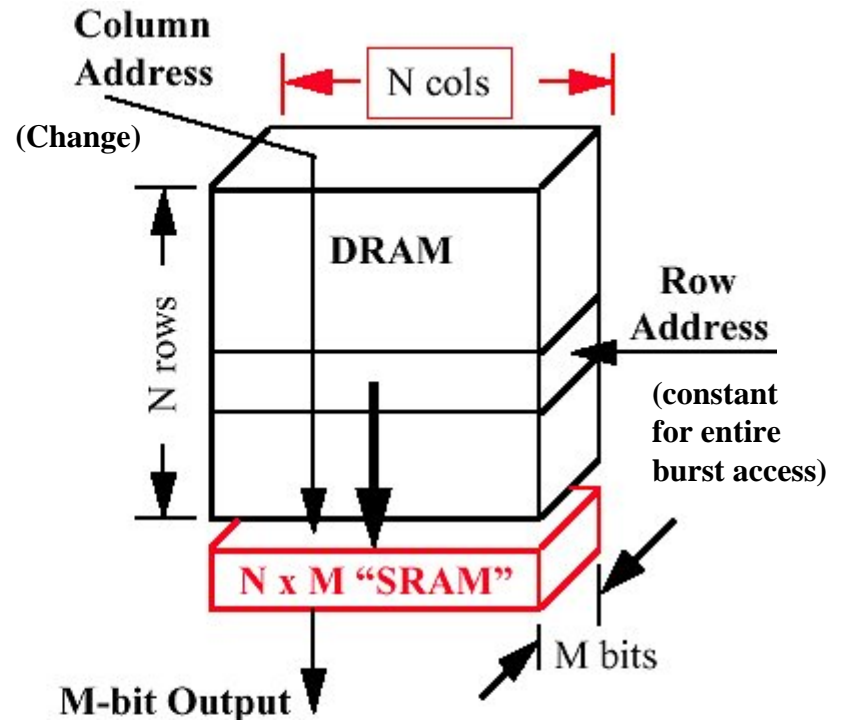
- 1 - Supply Row Address
- 2 - Supply Column Address
- 3 - Get Data

CMPE550 - Shaaban

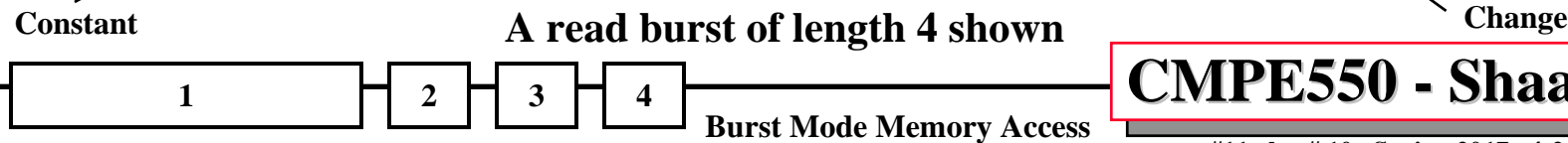
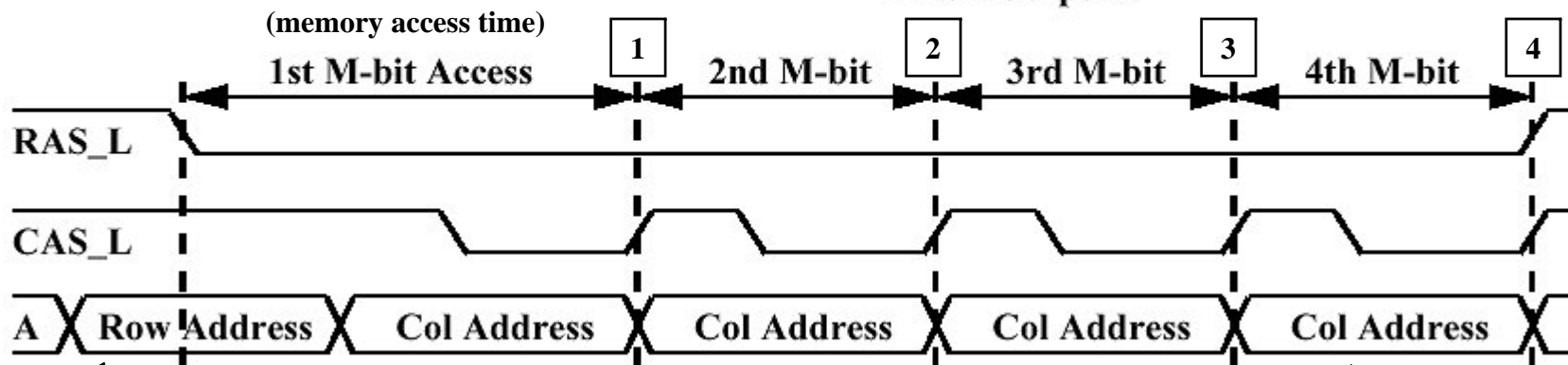
Fast Page Mode (FPM) DRAM (late 1980s)

Fast Page Mode DRAM

- N x M "SRAM" to save a row
- After a row is read into the register
 - Only CAS is needed to access other M-bit blocks on that row
 - RAS_L remains asserted while CAS_L is toggled



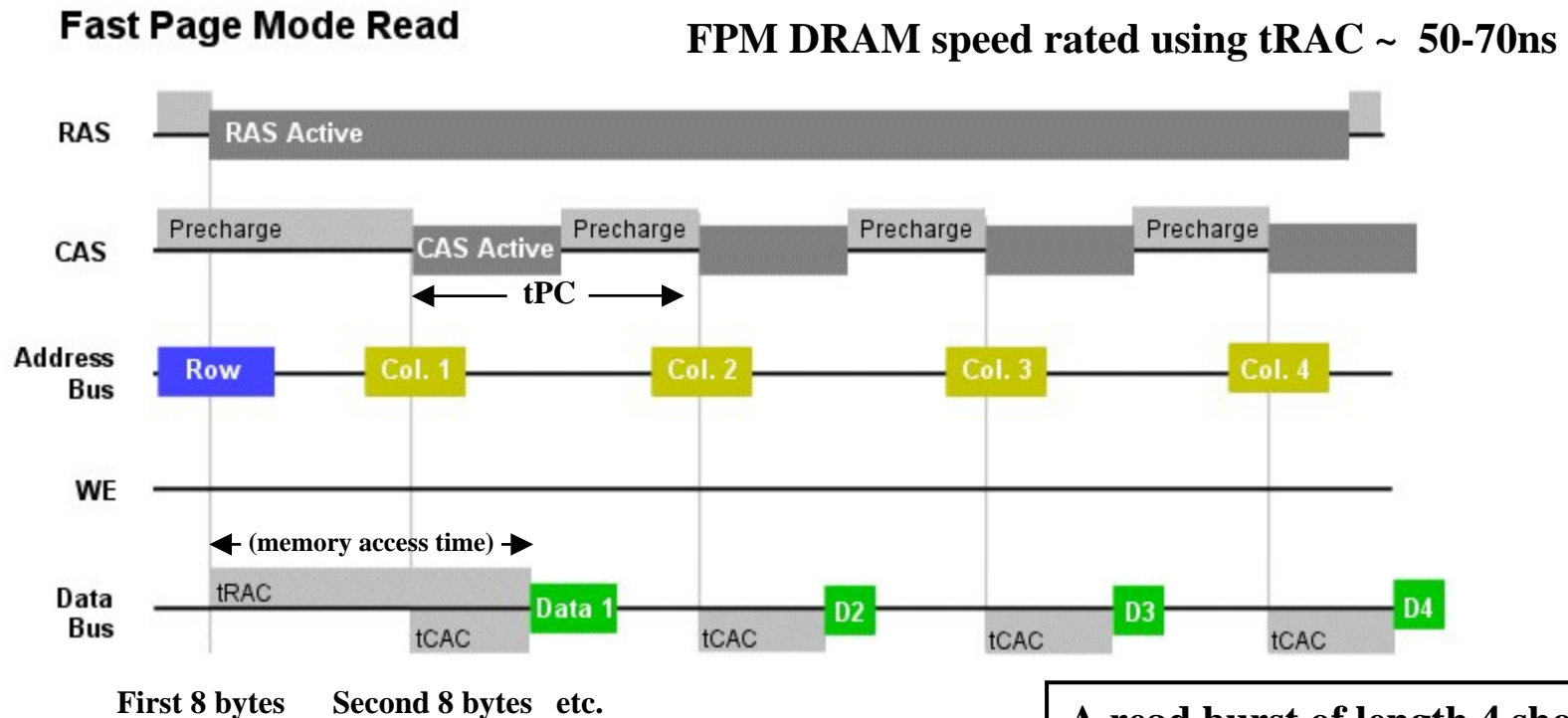
• The first "burst mode" DRAM



CMPE550 - Shaaban

Simplified Asynchronous Fast Page Mode (FPM) DRAM Read Timing

(late 80s)



A read burst of length 4 shown

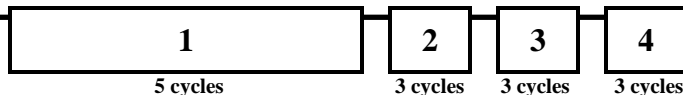
Typical timing at 66 MHz : 5-3-3-3 (burst of length 4)

For bus width = 64 bits = 8 bytes cache block size = 32 bytes

It takes = 5+3+3+3 = 14 memory cycles or 15 ns x 14 = 210 ns to read 32 byte block

Miss penalty for CPU running at 1 GHz = M = 15 x 14 = 210 CPU cycles

One memory cycle at 66 MHz = 1000/66 = 15 CPU cycles at 1 GHz

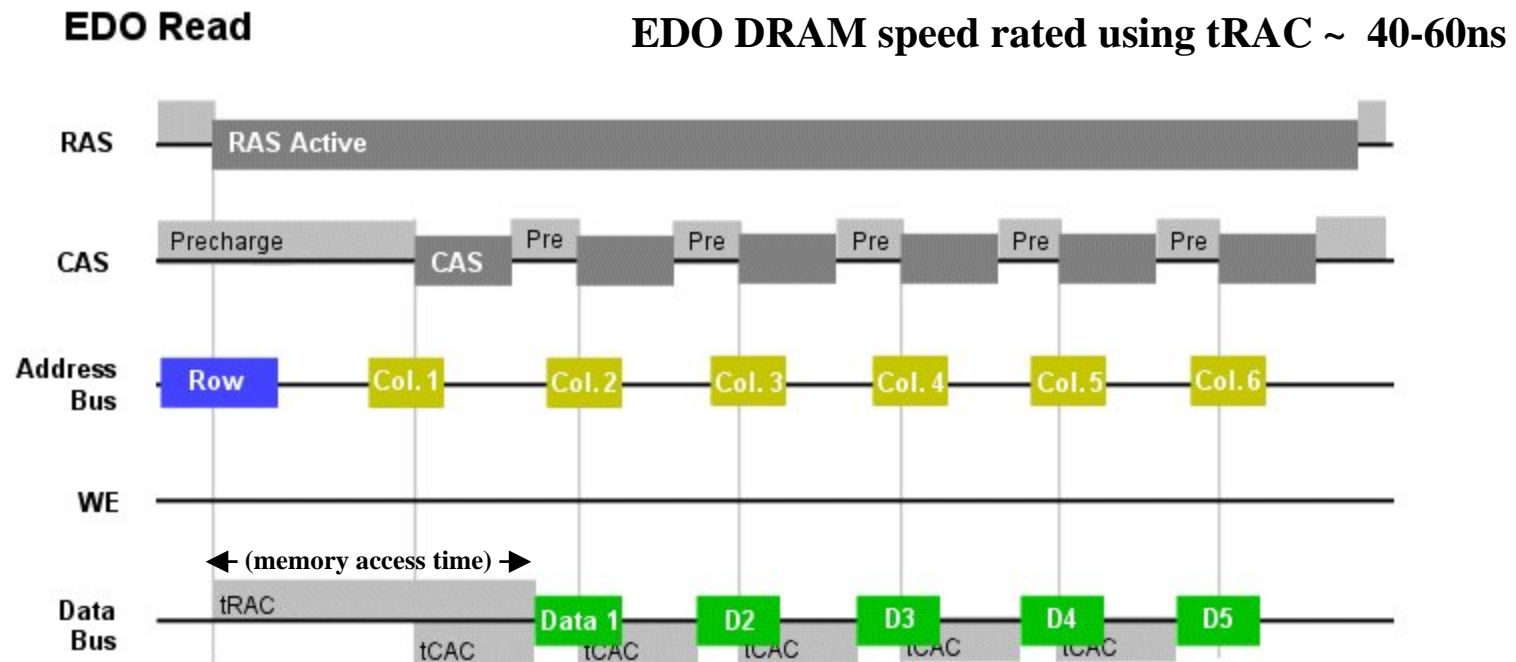


CMPE550 - Shaaban

Simplified Asynchronous Extended Data Out (EDO) DRAM Read Timing

(early 90s)

- Extended Data Out DRAM operates in a similar fashion to Fast Page Mode DRAM except putting data from one read on the output pins at the same time the column address for the next read is being latched in.



Typical timing at 66 MHz : 5-2-2-2 (burst of length 4)

For bus width = 64 bits = 8 bytes Max. Bandwidth = $8 \times 66 / 2 = 264$ Mbytes/sec

It takes = $5+2+2+2 = 11$ memory cycles or $15 \text{ ns} \times 11 = 165 \text{ ns}$ to read 32 byte cache block

Minimum Read Miss penalty for CPU running at 1 GHz = $M = 11 \times 15 = 165$ CPU cycles

One memory cycle at 66 MHz = $1000/66 = 15$ CPU cycles at 1 GHz

CMPE550 - Shaaban

Basic Memory Bandwidth Improvement/Miss Penalty (M) Latency Reduction Techniques

1

Wider Main Memory (CPU-Memory Bus/Interface):

wider FSB ?

Memory bus width is increased to a number of words (“usually” up to the size of a cache block).

- Memory bandwidth is proportional to memory bus width.
 - e.g Doubling the width of cache and memory doubles potential memory bandwidth available to the CPU. e.g 128 bit (16 bytes) memory bus instead of 64 bits (8 bytes) – now 24 bytes (192 bits)
- The miss penalty is reduced since fewer memory bus accesses are needed to fill a cache block on a miss.

2

Interleaved (Multi-Bank) Memory:

Memory is organized as a number of independent banks.

- Multiple interleaved memory reads or writes are accomplished by sending memory addresses to several memory banks at once or pipeline access to the banks.
- Interleaving factor: Refers to the mapping of memory addresses to memory banks. Goal reduce bank conflicts.

e.g. using 4 banks (width one word), bank 0 has all words whose address is:

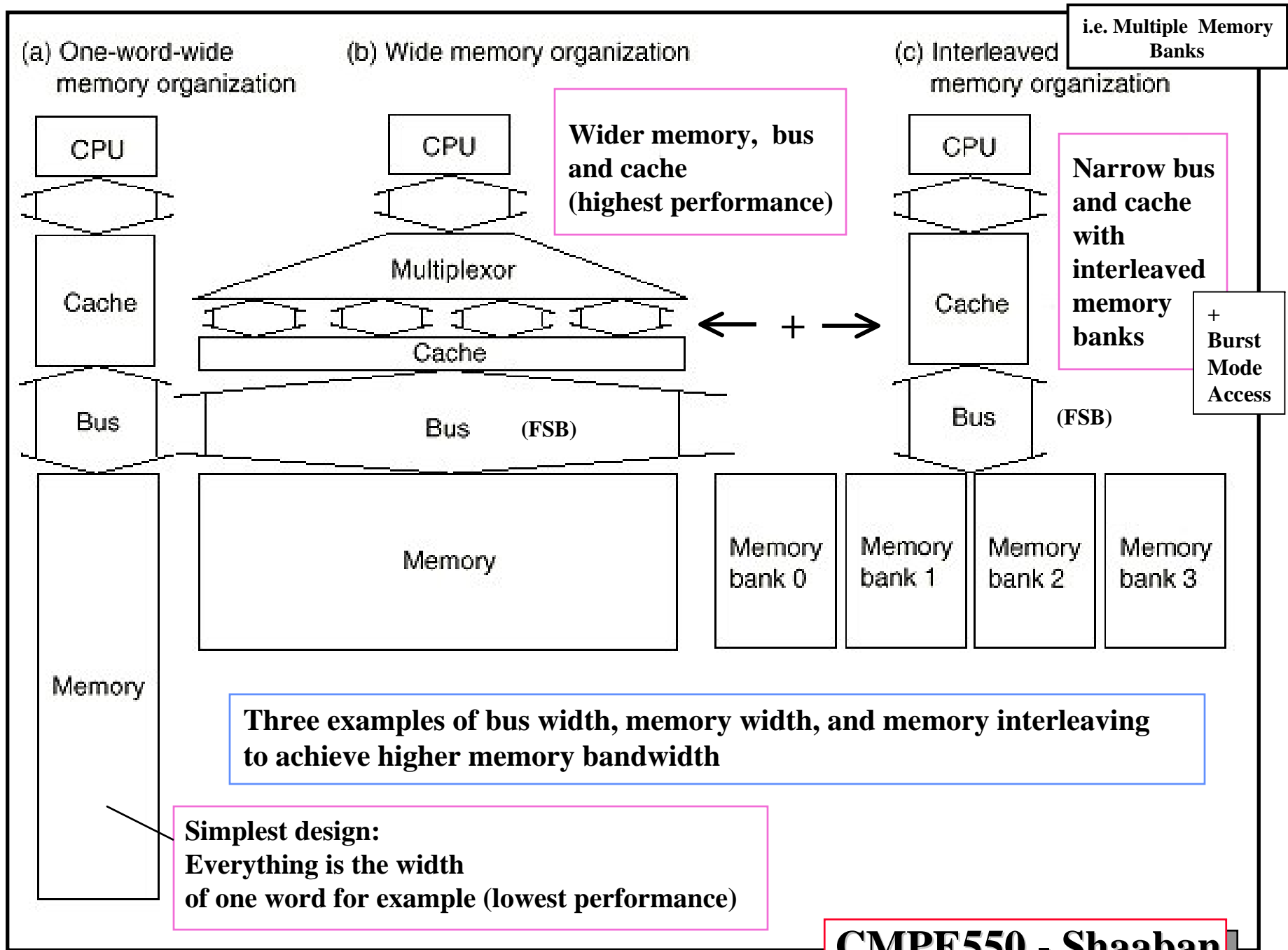
$$(\text{word address mod}) 4 = 0$$

3

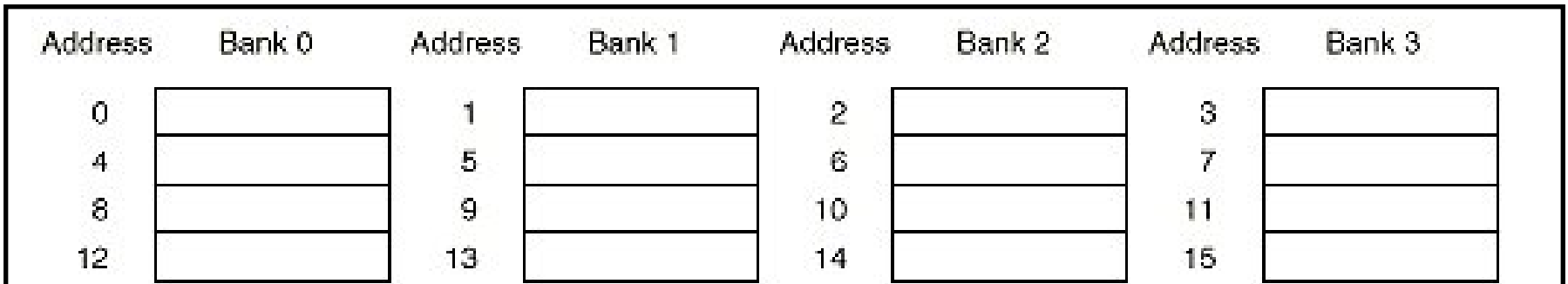
Burst Mode Memory Access

The above three techniques can also be applied to any cache level to reduce cache hit time and increase cache bandwidth.

CMPE550 - Shaaban



Front Side Bus (FSB) = System Bus = CPU-memory Bus



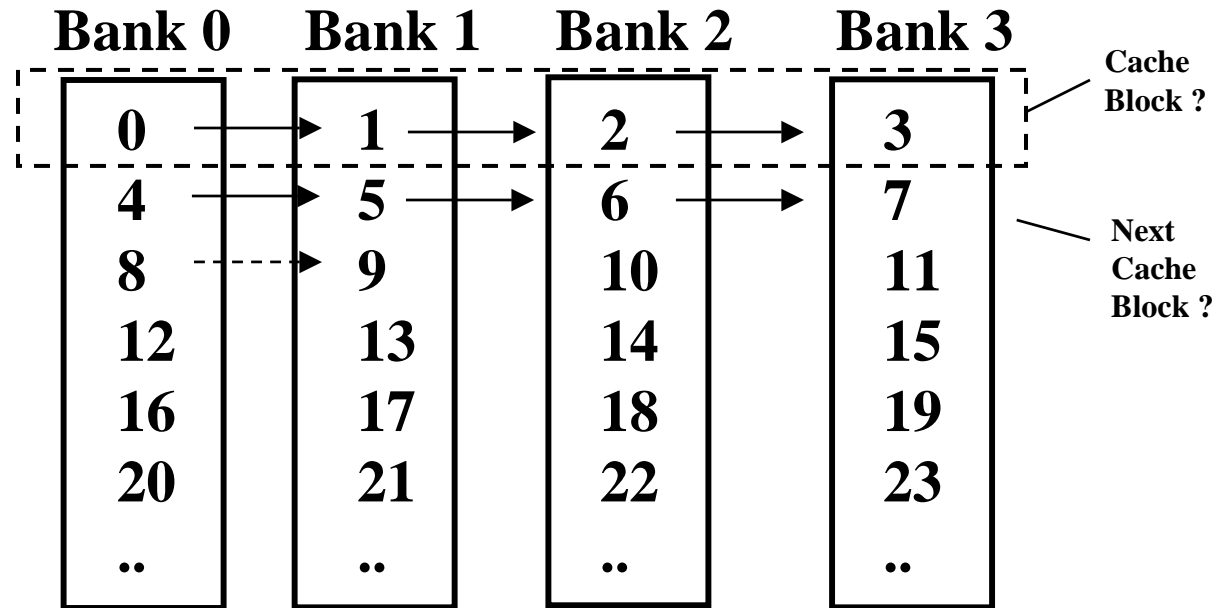
Four Way (Four Banks) Interleaved Memory

Memory Bank Number

Sequential Mapping of Memory Addresses To Memory Banks

Example

Address Within Bank



Bank Width = One Word

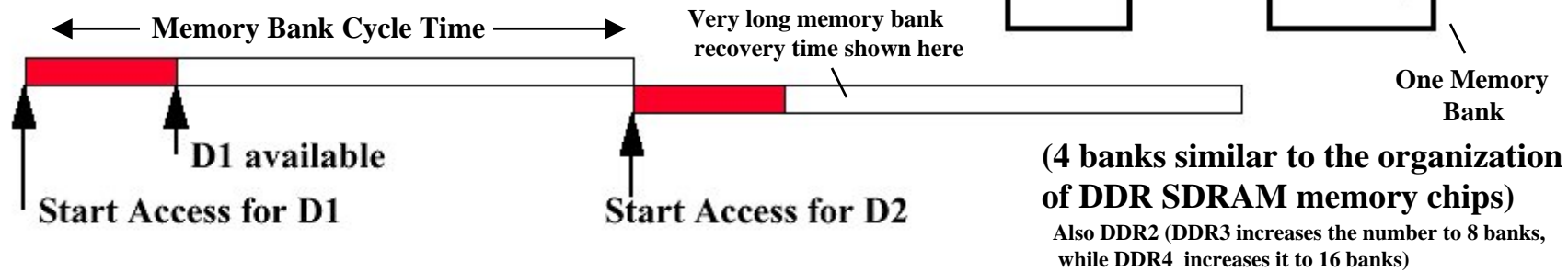
Bank Number = (Word Address) Mod (4)

CMPE550 - Shaaban

Memory Bank Interleaving (Multi-Banked Memory)

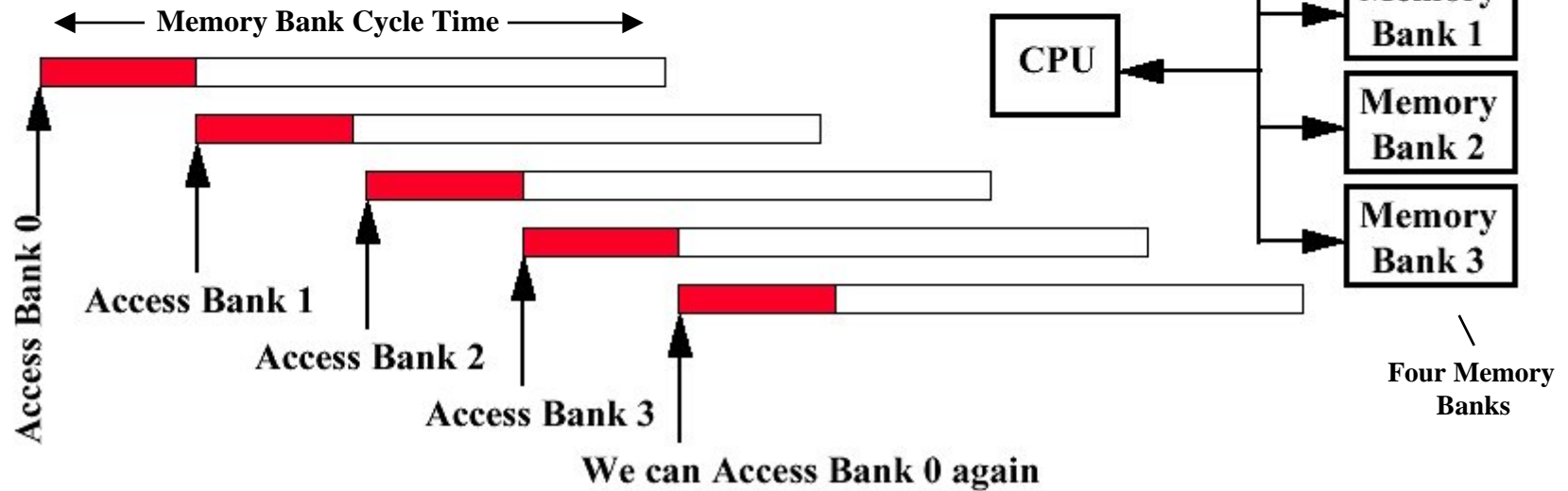
Can be applied at: 1- DRAM chip level (e.g. SDRAM, DDR) 2- DRAM module level 3- DRAM channel level

Access Pattern without Interleaving: (One Memory Bank)



Pipeline access to different memory banks to increase effective bandwidth

Access Pattern with 4-way Interleaving:



Number of banks \geq Number of cycles to access word in a bank

Bank interleaving does not reduce latency of accesses to the same bank

CMPE550 - Shaaban

Synchronous DRAM Generations Summary

All Use: 1- Fixed Clock Rate 2- Burst-Mode Access 3- Multiple Banks per DRAM chip

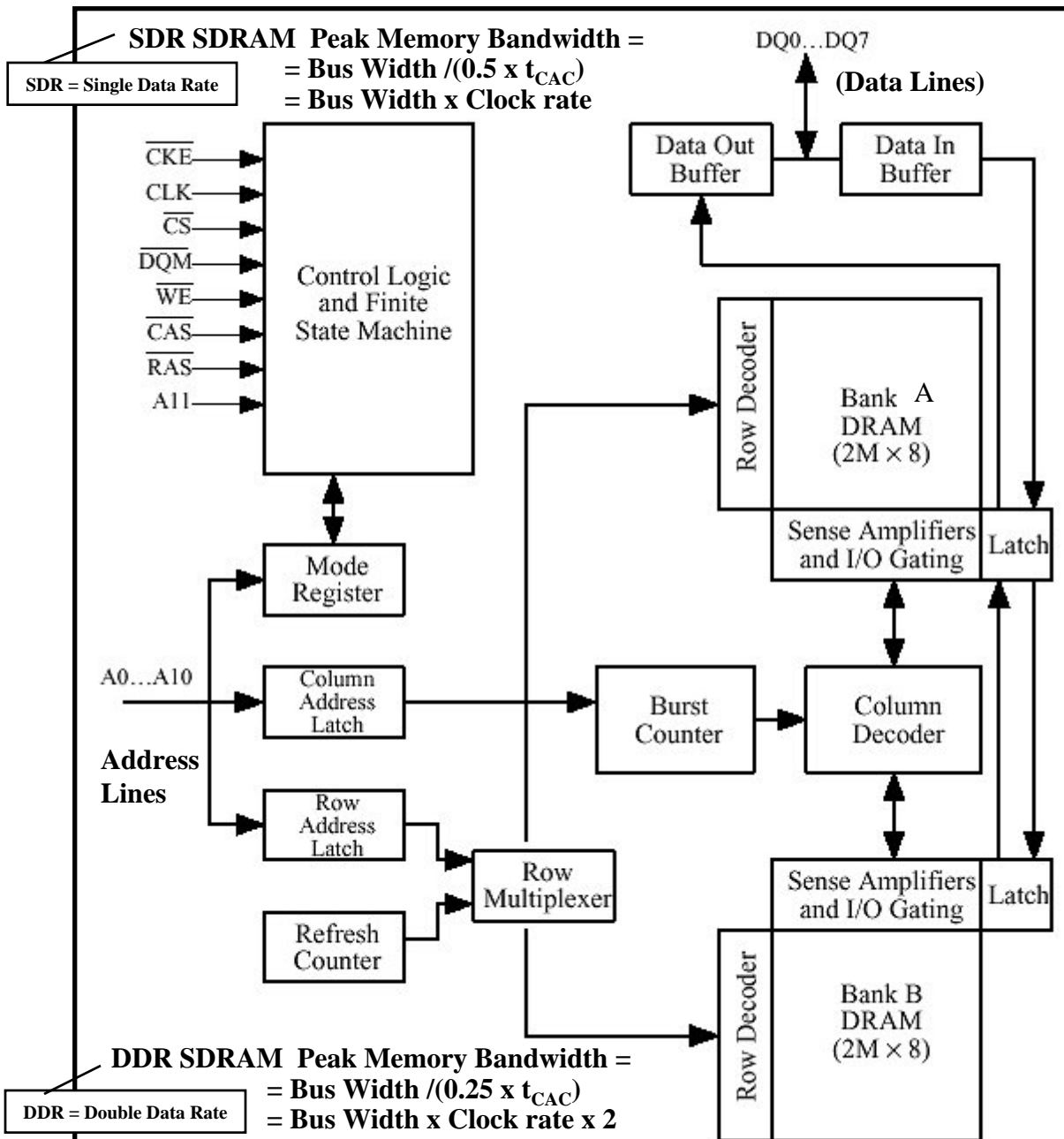
For Peak Bandwidth:
Initial burst latency not
taken into account

| | SDR (Single Data Rate) SDRAM | DDR (Double Data Rate) SDRAM | | | |
|----------------------------------------------------|------------------------------------|------------------------------|---------------------------|----------------------------|-----------------------------|
| | SDR | DDR | DDR2 | DDR3 | DDR4 |
| Year of Introduction | Late 1990's | 2002 | 2004 | 2007 | 2014 |
| # of Banks Per DRAM Chip | 2 | 4 | 4 | 8 | 16 |
| Example | PC100 | DDR400 (PC-3200) | DDR2-800 (PC2-6400) | DDR3-1600 (PC3-12800) | DDR4-3200 (PC4-25600) |
| Internal Base Frequency | 100 MHz | 200 MHz | 200 MHz | 200 MHz | 200 MHz |
| External Interface Frequency | 100 MHz | 200 MHz | 400 MHz | 800 MHz | 1600 MHz |
| Peak "Nominal" Bandwidth (per 8 byte module) | 0.8 GB/s (8 x 0.1) | 3.2 GB/s (8 x 0.2 x 2) | 6.4 GB/s (8 x 0.2 x 4) | 12.8 GB/s (8 x 0.2 x 8) | 25.6 GB/s (8 x 0.2 x 16) |
| Latency Range | 60-90 ns | 45-60 ns | 35-50 ns | 30-45 ns | 25-40 ns ? |

The latencies given only account for memory module latency and do not include memory controller latency or other address/data line delays. Thus realistic access latency is longer

CMPE550 - Shaaban

All synchronous memory types above use burst-mode access with multiple memory banks per DRAM chip



SDR SDRAM Peak Memory Bandwidth =
 = Bus Width / (0.5 x t_{CAC})
 = Bus Width x Clock rate

SDR = Single Data Rate

Synchronous Dynamic RAM, (SDR SDRAM) Organization

(mid 90s)

SDRAM speed is rated at max. clock speed supported:
 100MHZ = PC100
 133MHZ = PC133

SDR = Single Data Rate

DDR SDRAM

DDR = Double Data Rate

(late 90s - 2006)

organization is similar but **four banks** are used in each DDR SDRAM chip instead of two.

Also DDR2

(DDR3 increases the number of banks to 8 banks)

Data transfer on both **rising and falling edges of the clock**

DDR SDRAM rated by maximum or peak memory bandwidth
 PC3200 = 8 bytes x 200 MHz x 2
 = 3200 Mbytes/sec

| | | | | | | | |
|---|---|---|---|-------|---|---|-----------|
| 1 | 2 | 3 | 4 | SDRAM | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | DDR SDRAM |

Timing Comparison

CMPE550 - Shaaban

Comparison of Synchronous Dynamic RAM SDRAM Generations:

DDR2 Vs. DDR and SDR SDRAM

For DDR3: The trend continues with another external frequency doubling

Single Data Rate (SDR) SDRAM transfers data on every rising edge of the clock.

Whereas both DDR and DDR2 are double pumped; they transfer data on the rising and falling edges of the clock.

DDR2 vs. DDR:

- **DDR2 doubles bus frequency** for the same physical DRAM chip clock rate (as shown), thus **doubling the effective data rate another time**.

- Ability for much higher clock speeds than DDR, due to design improvements (still 4-banks per chip):

- DDR2's bus frequency is boosted by electrical interface improvements, on-die termination, prefetch buffers and off-chip drivers.

- **However, latency vs. DDR is greatly increased as a trade-off.**

Internal Base Frequency = 133 MHz

Peak bandwidth given for a single 64bit memory channel (i.e 8-byte memory bus width)

4258 MB/s
= 8 x 133 x 4

DDR2
SDRAM

Shown: DDR2-533 (PC2-4200)
~ 4.2 GB/s peak bandwidth

2128 MB/s
= 8 x 133 x 2

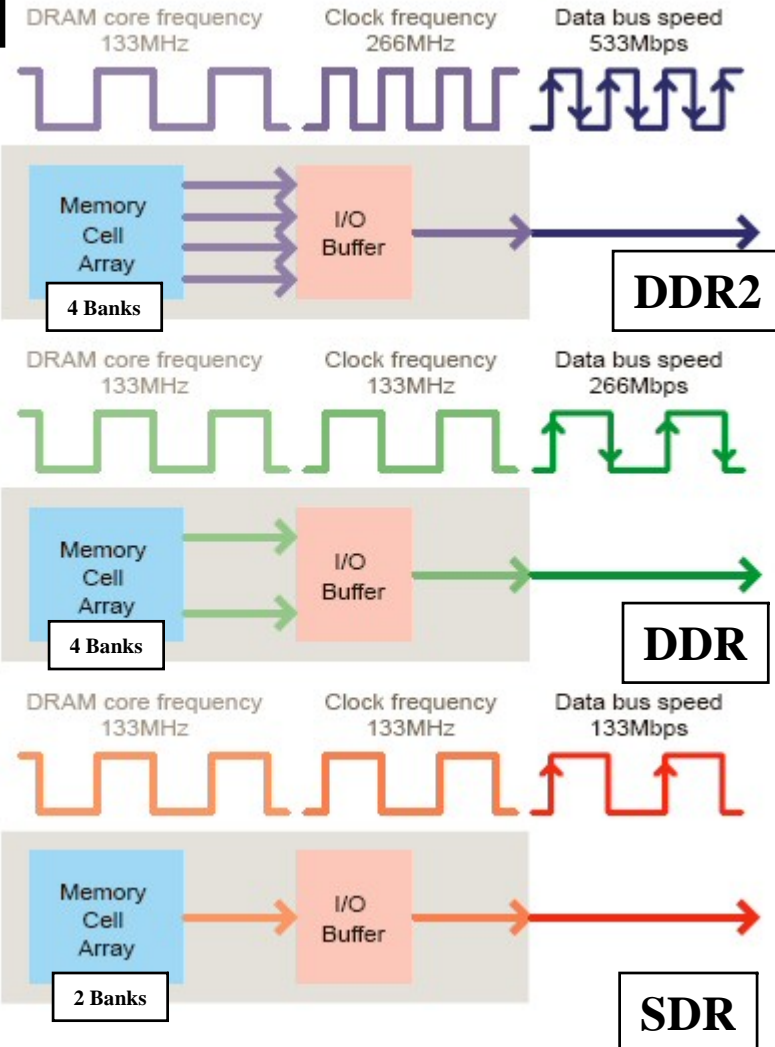
DDR
SDRAM

Shown: DDR-266 (PC-2100)
~ 2.1 GB/s peak bandwidth

1064 MB/s
= 8 x 133

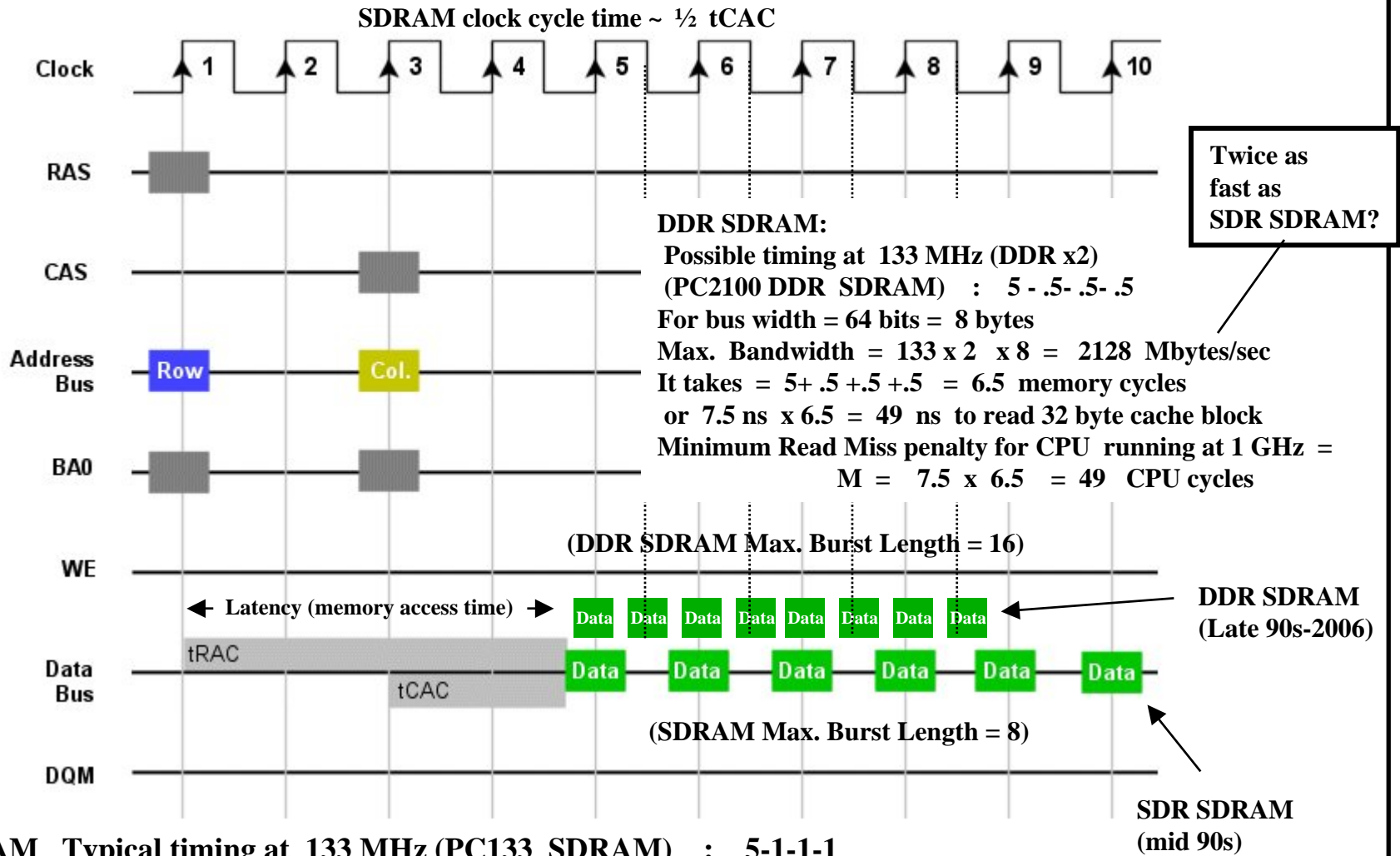
SDR
SDRAM

Shown: PC133
~ 1.05 GB/s peak bandwidth



CMPE550 - Shaaban

SDRAM Read Simplified SDR SDRAM/DDR SDRAM Read Timing



SDR

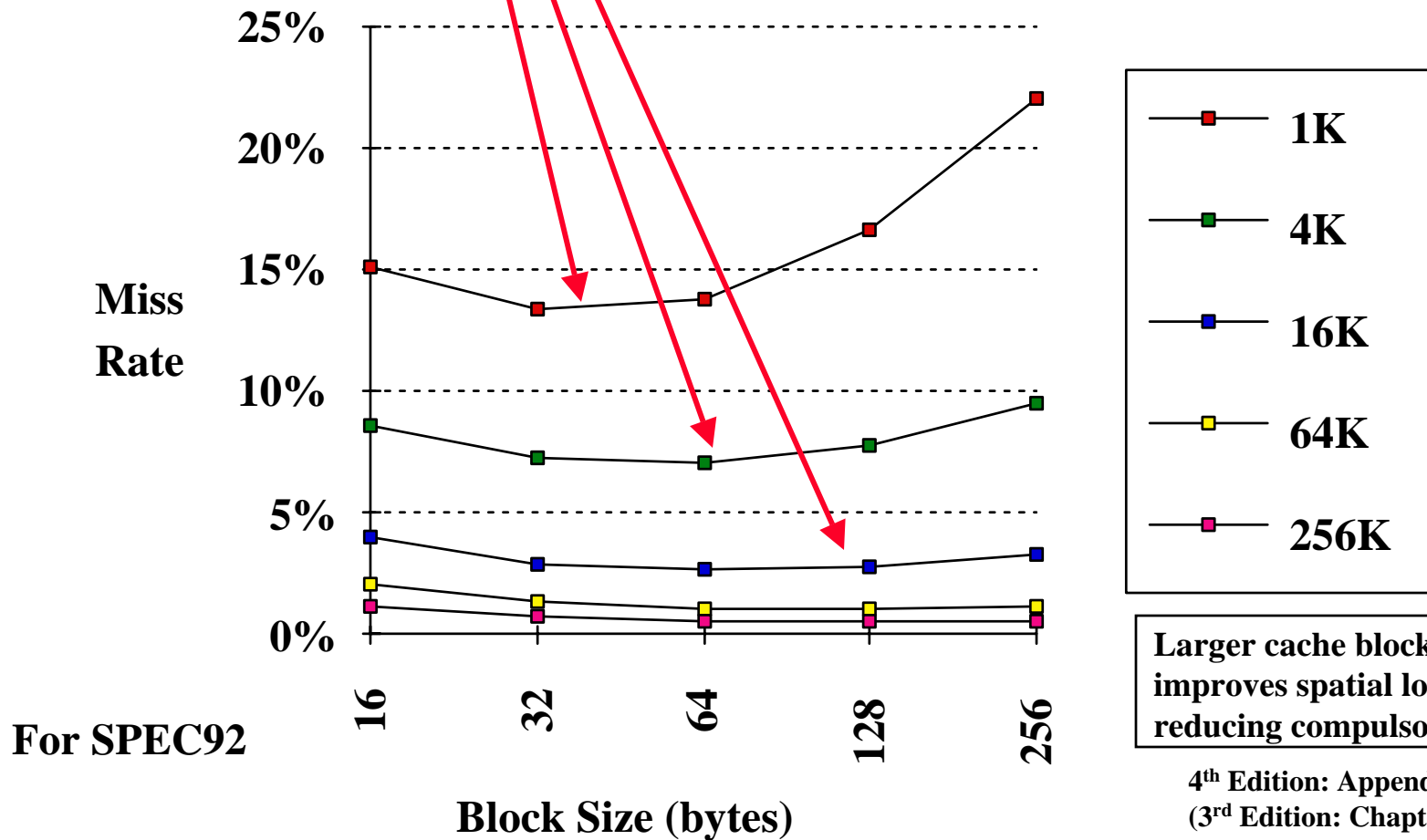
In this example for SDR SDRAM: M = 60 cycles for DDR SDRAM: M = 49 cycles
 Thus accounting for access latency DDR is 60/49 = 1.22 times faster
 Not twice as fast (2128/1064 = 2) as indicated by peak bandwidth!

CMPE550 - Shaaban

The Impact of Larger Cache Block Size on Miss Rate

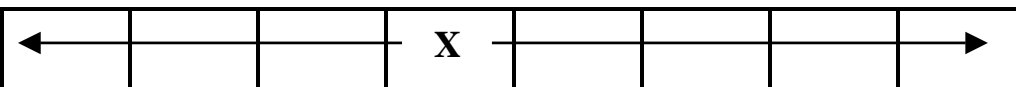
- A larger cache block size improves cache performance by taking better advantage of spatial locality. However, for a fixed cache size, larger block sizes mean fewer cache block frames.

Performance keeps improving to a limit when the fewer number of cache block frames increases conflicts and thus overall cache miss rate.



Larger cache block size improves spatial locality reducing compulsory misses

4th Edition: Appendix C.3
(3rd Edition: Chapter 5.5)



CMPE550 - Shaaban

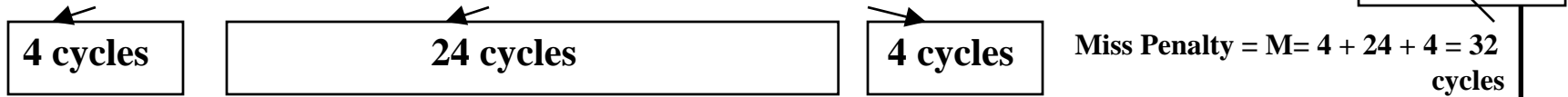
Memory Width, Interleaving: Performance Example

(i.e Multiple Memory banks + Burst Mode Access)

Given the following system parameters with single unified cache level L₁ (ignoring write policy):

Block size= 1 word Memory bus width= 1 word Miss rate =3% M = Miss penalty = 32 cycles

(4 cycles to send address 24 cycles access time, 4 cycles to send a word to CPU)



Memory access/instruction = 1.2 CPI_{execution} (ignoring cache misses) = 2

Miss rate (block size = 2 word = 8 bytes) = 2% Miss rate (block size = 4 words = 16 bytes) = 1%

- The CPI of the base machine with 1-word blocks = $2 + (1.2 \times 0.03 \times 32) = 3.15$ (For Base system)

Increasing the block size to two words (64 bits) gives the following CPI: (miss rate = 2%)

- 32-bit bus and memory, no interleaving, $M = 2 \times 32 = 64$ cycles $CPI = 2 + (1.2 \times .02 \times 64) = 3.54$
- 32-bit bus and memory, interleaved, $M = 4 + 24 + 8 = 36$ cycles $CPI = 2 + (1.2 \times .02 \times 36) = 2.86$
- 64-bit bus and memory, no interleaving, $M = 32$ cycles $CPI = 2 + (1.2 \times 0.02 \times 32) = 2.77$

Increasing the block size to four words (128 bits); resulting CPI: (miss rate = 1%)

- 32-bit bus and memory, no interleaving, $M = 4 \times 32 = 128$ cycles $CPI = 2 + (1.2 \times 0.01 \times 128) = 3.54$
- 32-bit bus and memory, interleaved, $M = 4 + 24 + 16 = 44$ cycles $CPI = 2 + (1.2 \times 0.01 \times 44) = 2.53$
- 64-bit bus and memory, no interleaving, $M = 2 \times 32 = 64$ cycles $CPI = 2 + (1.2 \times 0.01 \times 64) = 2.77$
- 64-bit bus and memory, interleaved, $M = 4 + 24 + 8 = 36$ cycles $CPI = 2 + (1.2 \times 0.01 \times 36) = 2.43$
- 128-bit bus and memory, no interleaving, $M = 32$ cycles $CPI = 2 + (1.2 \times 0.01 \times 32) = 2.38$



CMPE550 - Shaaban

Miss Penalty = M = Number of CPU stall cycles for an access missed in cache and satisfied by main memory

Three-Level Cache Example

- CPU with $CPI_{\text{execution}} = 1.1$ running at clock rate = 500 MHz
- 1.3 memory accesses per instruction.
- L_1 cache operates at 500 MHz (no stalls on a hit in L_1) with a miss rate of 5%
- L_2 hit access time = 3 cycles ($T_2 = 2$ stall cycles per hit), local miss rate 40%
- L_3 hit access time = 6 cycles ($T_3 = 5$ stall cycles per hit), local miss rate 50%,
- Memory access penalty, $M = 100$ cycles (stall cycles per access). Find CPI.

All Unified
Ignoring write policy

With No Cache, $CPI = 1.1 + 1.3 \times 100 = 131.1$

With single L_1 , $CPI = 1.1 + 1.3 \times .05 \times 100 = 7.6$

With L_1, L_2 $CPI = 1.1 + 1.3 \times (.05 \times .6 \times 2 + .05 \times .4 \times 100) = 3.778$

$$CPI = CPI_{\text{execution}} + \text{Mem Stall cycles per instruction}$$

Mem Stall cycles per instruction = Mem accesses per instruction \times Stall cycles per access

$$\begin{aligned} \text{Stall cycles per memory access} &= (1-H_1) \times H_2 \times T_2 + (1-H_1) \times (1-H_2) \times H_3 \times T_3 + (1-H_1)(1-H_2)(1-H_3) \times M \\ &= .05 \times .6 \times 2 + .05 \times .4 \times .5 \times 5 + .05 \times .4 \times .5 \times 100 \\ &= .06 + .05 + 1 = 1.11 \end{aligned}$$

AMAT = 1.11 + 1 = 2.11 cycles (vs. AMAT = 3.06 with L_1, L_2 , vs. 5 with L_1 only)

$CPI = 1.1 + 1.3 \times 1.11 = 2.54$ With L_1, L_2, L_3

Speedup compared to L_1 only = $7.6/2.54 = 3$

Speedup compared to L_1, L_2 = $3.778/2.54 = 1.49$

Repeated here from lecture 8

CMPE550 - Shaaban

Memory Access Tree For Example

3-Level (All Unified) Cache Performance

Memory Access Tree (Ignoring Write Policy)

CPU Stall Cycles Per Memory Access

CPU Memory Access (100%)

$$\text{CPI} = \text{CPI}_{\text{execution}} + (1 + \text{fraction of loads and stores}) \times \text{stalls per access}$$

$$\text{CPI} = 1.1 + 1.3 \times 1.11 = 2.54$$

H1 = .95 or 95%

L1 Hit:
 Hit Access Time = 1
 Stalls Per access = 0
 Stalls = H1 x 0 = 0
 (No Stall)

L1 Miss:
 % = (1-H1) = .05 or 5%

H1 = 95% T1 = 0 cycles
 H2 = 60% T2 = 2 cycles
 H3 = 50% T3 = 5 cycles
 M = 100 cycles

Stalls on a hit

$$(1-H1) \times H2$$

$$= .05 \times .6$$

$$= .03 \text{ or } 3\%$$

L1 Miss, L2 Hit:
 Hit Access Time = T2 + 1 = 3
 Stalls per L2 Hit = T2 = 2
 Stalls = (1-H1) x H2 x T2
 = .05 x .6 x 2 = .06

L1 Miss, L2 Miss:
 % = (1-H1)(1-H2) = .05 x .4 = .02 or 2%

$$(1-H1) \times (1-H2) \times H3$$

$$= .05 \times .4 \times .5$$

$$= .01 \text{ or } 1\%$$

$$(1-H1)(1-H2)(1-H3)$$

$$= .05 \times .4 \times .5$$

$$= .01 \text{ or } 1\%$$

Full Miss

L3 Hit:
 Hit Access Time = T3 + 1 = 6
 Stalls per L2 Hit = T3 = 5
 Stalls = (1-H1) x (1-H2) x H3 x T3
 = .01 x 5 = .05 cycles

L1 Miss, L2, Miss, L3 Miss:
 Miss Penalty = M = 100
 Stalls = (1-H1)(1-H2)(1-H3) x M
 = .01 x 100 = 1 cycle

$$\text{Stall cycles per memory access} = (1-H1) \times H2 \times T2 + (1-H1) \times (1-H2) \times H3 \times T3 + (1-H1)(1-H2)(1-H3) \times M$$

$$= .06 + .05 + 1 = 1.11$$

$$\text{AMAT} = 1 + \text{Stall cycles per memory access}$$

$$= 1 + 1.11 = 2.11 \text{ cycles}$$

T2 = 2 cycles = Stalls per hit access for Level 2
 T3 = 5 cycles = Stalls per hit access for Level 3
 M = Memory Miss Penalty = M = 100 cycles

Repeated here from lecture 8

CMPE550 - Shaaban

Program Steady-State Bandwidth-Usage Example

- In the previous example with three levels of cache (all unified, ignore write policy)
- CPU with $CPI_{\text{execution}} = 1.1$ running at clock rate = 500 MHz
- 1.3 memory accesses per instruction.
- L_1 cache operates at 500 MHz (no stalls on a hit in L_1) with a miss rate of 5%
- L_2 hit access time = 3 cycles ($T_2 = 2$ stall cycles per hit), local miss rate 40%
- L_3 hit access time = 6 cycles ($T_3 = 5$ stall cycles per hit), local miss rate 50%,
- Memory access penalty, $M = 100$ cycles (stall cycles per access to deliver 32 bytes from main memory to CPU)
- We found the CPI:
 - With No Cache, $CPI = 1.1 + 1.3 \times 100 = 131.1$
 - With single L_1 , $CPI = 1.1 + 1.3 \times .05 \times 100 = 7.6$
 - With L_1, L_2 $CPI = 1.1 + 1.3 \times (.05 \times .6 \times 2 + .05 \times .4 \times 100) = 3.778$
 - With L_1, L_2, L_3 $CPI = 1.1 + 1.3 \times 1.11 = 2.54$

Assuming that all cache blocks are 32 bytes

For each of the three cases with cache:

i.e. L_1 only, L_1 and L_2 , all three levels

- A. What is the peak (or maximum) number of memory accesses and effective peak bandwidth for each cache level and main memory?
- B. What is the total number of memory accesses generated by the CPU per second?
- C. What percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

Program Steady-State Bandwidth-Usage Example

A. What is the peak (or maximum) number of memory accesses and effective peak bandwidth for each cache level and main memory?

- L1 cache requires 1 CPU cycle to deliver 32 bytes, thus:

Maximum L1 accesses per second = 500×10^6 accesses/second

Maximum effective L1 bandwidth = $32 \times 500 \times 10^6 = 16,000 \times 10^6 = 16 \times 10^9$ bytes/sec
Cache Block Size

- L2 cache requires 3 CPU cycles to deliver 32 bytes, thus:

Maximum L2 accesses per second = $500/3 \times 10^6 = 166.67 \times 10^6$ accesses/second

Maximum effective L2 bandwidth = $32 \times 166.67 \times 10^6 = 5,333.33 \times 10^6 = 5.33 \times 10^9$ bytes/sec

- L3 cache requires 6 CPU cycles to deliver 32 bytes, thus:

Maximum L3 accesses per second = $500/6 \times 10^6 = 83.33 \times 10^6$ accesses/second

Maximum effective L3 bandwidth = $32 \times 83.33 \times 10^6 = 2,666.67 \times 10^6 = 2.67 \times 10^9$ bytes/sec

- Memory requires 101 CPU cycles ($101 = M+1 = 100+1$) to deliver 32 bytes, thus:

Maximum main memory accesses per second = $500/101 \times 10^6 = 4.95 \times 10^6$ accesses/second

Maximum effective main memory bandwidth = $32 \times 4.95 \times 10^6 = 158.42 \times 10^6$ bytes/sec

Cache block size = 32 bytes

CMPE550 - Shaaban

Program Steady-State Bandwidth-Usage Example

- For CPU with L1 Cache:

B. What is the total number of memory accesses generated by the CPU per second?

- The total number of memory accesses generated by the CPU per second = (memory access/instruction) x clock rate / CPI = $1.3 \times 500 \times 10^6 / \text{CPI} = 650 \times 10^6 / \text{CPI}$
- With single L1 cache CPI was found = 7.6
– CPU memory accesses = $650 \times 10^6 / 7.6 = 85 \times 10^6$ accesses/sec

C. What percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

- For L1:

The percentage of CPU memory accesses that reach L1 = 100%

L1 Cache bandwidth usage = $32 \times 85 \times 10^6 = 2,720 \times 10^6 = 2.7 \times 10^9$ bytes/sec

Percentage of L1 bandwidth used = $2,720 / 16,000 = 0.17$ or 17%

(or by just dividing CPU accesses / peak L1 accesses = $85/500 = 0.17 = 17\%$)

- For Main Memory:

The percentage of CPU memory accesses that reach main memory = $(1-H1) = 0.05$ or 5%

Main memory bandwidth usage = $0.05 \times 32 \times 85 \times 10^6 = 136 \times 10^6$ bytes/sec

Percentage of main memory bandwidth used = $136 / 158.42 = 0.8585$ or 85.85%

Program Steady-State Bandwidth-Usage Example

- For CPU with L1, L2 Cache:

B. What is the total number of memory accesses generated by the CPU per second?

- The total number of memory accesses generated by the CPU per second = (memory access/instruction) x clock rate / CPI = $1.3 \times 500 \times 10^6 / \text{CPI} = 650 \times 10^6 / \text{CPI}$

- With L1, L2 cache CPI was found = 3.778

– CPU memory accesses = $650 \times 10^6 / 3.778 = 172 \times 10^6$ accesses/sec Vs. With L1 only = 85×10^6 accesses/sec

C. What percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

- For L1:

The percentage of CPU memory accesses that reach L1 = 100%

L1 Cache bandwidth usage = $32 \times 172 \times 10^6 = 5,505 \times 10^6 = 5.505 \times 10^9$ bytes/sec

Percentage of L1 bandwidth used = $5,505 / 16,000 = 0.344$ or 34.4% Vs. With L1 only = 17%

(or by just dividing CPU accesses / peak L1 accesses = $172/500 = 0.344 = 34.4\%$)

- For L2:

The percentage of CPU memory accesses that reach L2 = (I-H1) = 0.05 or 5%

L2 Cache bandwidth usage = $0.05 \times 32 \times 172 \times 10^6 = 275.28 \times 10^6$ bytes/sec

Percentage of L2 bandwidth used = $275.28 / 5,333.33 = 0.0516$ or 5.16%

(or by just dividing CPU accesses that reach L2 / peak L2 accesses = $0.05 \times 172 / 166.67 = 8.6 / 166.67 = 0.0516 = 5.16\%$)

- For Main Memory:

The percentage of CPU memory accesses that reach main memory = (1-H1) x (1-H2) = $0.05 \times 0.4 = 0.02$ or 2%

Main memory bandwidth usage = $0.02 \times 32 \times 172 \times 10^6 = 110.11 \times 10^6$ bytes/sec

Percentage of main memory bandwidth used = $110.11 / 158.42 = 0.695$ or 69.5% Vs. With L1 only = 85.5%

Exercises: What if Level 1 (L1) is split?
What if Level 2 (L2) is write back with write allocate?

CMPE550 - Shaaban

Program Steady-State Bandwidth-Usage Example

- For CPU with L1, L2, L3 Cache:

B. What is the total number of memory accesses generated by the CPU per second?

- The total number of memory accesses generated by the CPU per second =
(memory access/instruction) x clock rate / CPI = $1.3 \times 500 \times 10^6 / \text{CPI} = 650 \times 10^6 / \text{CPI}$
- With L1, L2, L3 cache CPI was found = 2.54
 – CPU memory accesses = $650 \times 10^6 / 2.54 = 255.9 \times 10^6$ accesses/sec

Vs. With L1 only = 85×10^6 accesses/sec
 With L1, L2 = 172×10^6 accesses/sec

C. What percentage of these memory accesses reach each cache level/memory and what percentage of each cache level/memory bandwidth is used by the CPU?

- For L1:

The percentage of CPU memory accesses that reach L1 = 100%

L1 Cache bandwidth usage = $32 \times 255.9 \times 10^6 = 8,188 \times 10^6 = 8.188 \times 10^9$ bytes/sec

Percentage of L1 bandwidth used = $8,188 / 16,000 = 0.5118$ or 51.18%

(or by just dividing CPU accesses / peak L1 accesses = $172/500 = 0.344 = 34.4\%$)

Vs. With L1 only = 17%
 With L1, L2 = 34.4%

- For L2:

The percentage of CPU memory accesses that reach L2 = $(1-H1) = 0.05$ or 5%

L2 Cache bandwidth usage = $0.05 \times 32 \times 255.9 \times 10^6 = 409.45 \times 10^6$ bytes/sec

Percentage of L2 bandwidth used = $409.45 / 5,333.33 = 0.077$ or 7.7 %

(or by just dividing CPU accesses that reach L2 / peak L2 accesses = $0.05 \times 255.9 / 166.67 = 12.795 / 166.67 = 0.077 = 7.7\%$)

Vs. With L1, L2 only = 5.16%

- For L3:

The percentage of CPU memory accesses that reach L2 = $(1-H1) \times (1-H2) = 0.02$ or 2%

L3 Cache bandwidth usage = $0.02 \times 32 \times 255.9 \times 10^6 = 163.78 \times 10^6$ bytes/sec

Percentage of L3 bandwidth used = $163.78 / 2,666.67 = 0.061$ or 6.1 %

(or by just dividing CPU accesses that reach L3 / peak L3 accesses = $0.02 \times 255.9 / 83.33 = 5.118 / 83.33 = 0.061 = 6.1\%$)

- For Main Memory:

The percentage of CPU memory accesses that reach main memory = $(1-H1) \times (1-H2) \times (1-H3) = .05 \times .4 \times .5 = 0.01$ or 1%

Main memory bandwidth usage = $0.01 \times 32 \times 255.9 \times 10^6 = 81.89 \times 10^6$ bytes/sec

Percentage of main memory bandwidth used = $110.11 / 158.42 = 0.517$ or 51.7%

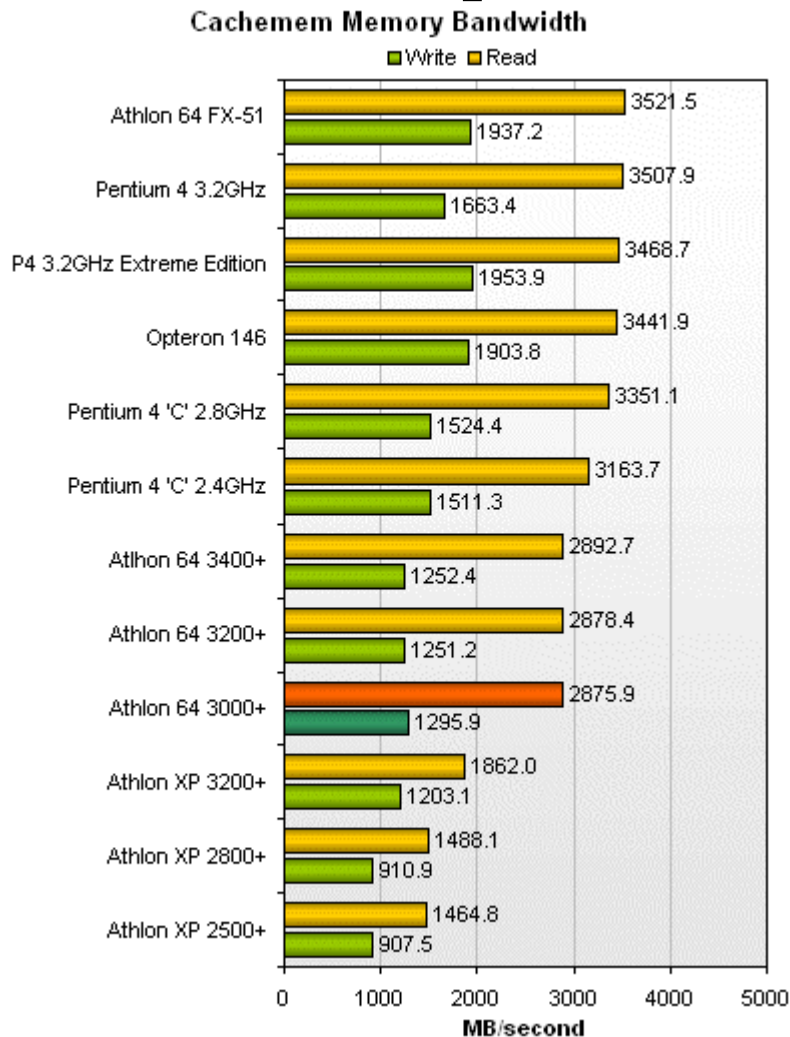
Vs. With L1 only = 85.5%
 With L1, L2 = 69.5%

Exercises: What if Level 1 (L1) is split?
 What if Level 3 (L3) is write back with write allocate?

CMPE550 - Shaaban

X86 CPU Dual Channel PC3200 DDR SDRAM

Sample (Realistic?) Bandwidth Data



**Dual (64-bit) Channel PC3200 DDR SDRAM
has a theoretical peak bandwidth of**

$$400 \text{ MHz} \times 8 \text{ bytes} \times 2 = 6400 \text{ MB/s}$$

Is memory bandwidth still an issue?

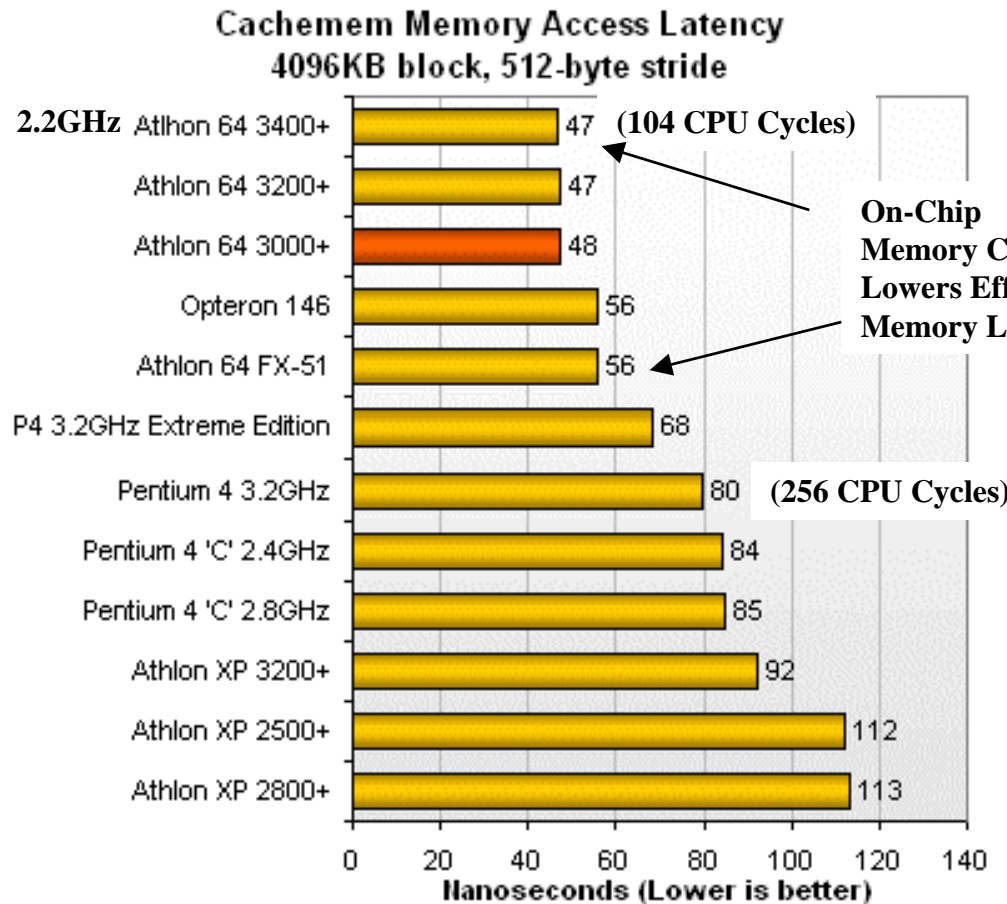
Source: The Tech Report 1-21-2004

<http://www.tech-report.com/reviews/2004q1/athlon64-3000/index.x?pg=3>

CMPE550 - Shaaban

X86 CPU Dual Channel PC3200 DDR SDRAM

Sample (Realistic?) Latency Data



PC3200 DDR SDRAM

has a theoretical latency range of
18-40 ns

(not accounting for memory controller
latency or other address/data line delays).

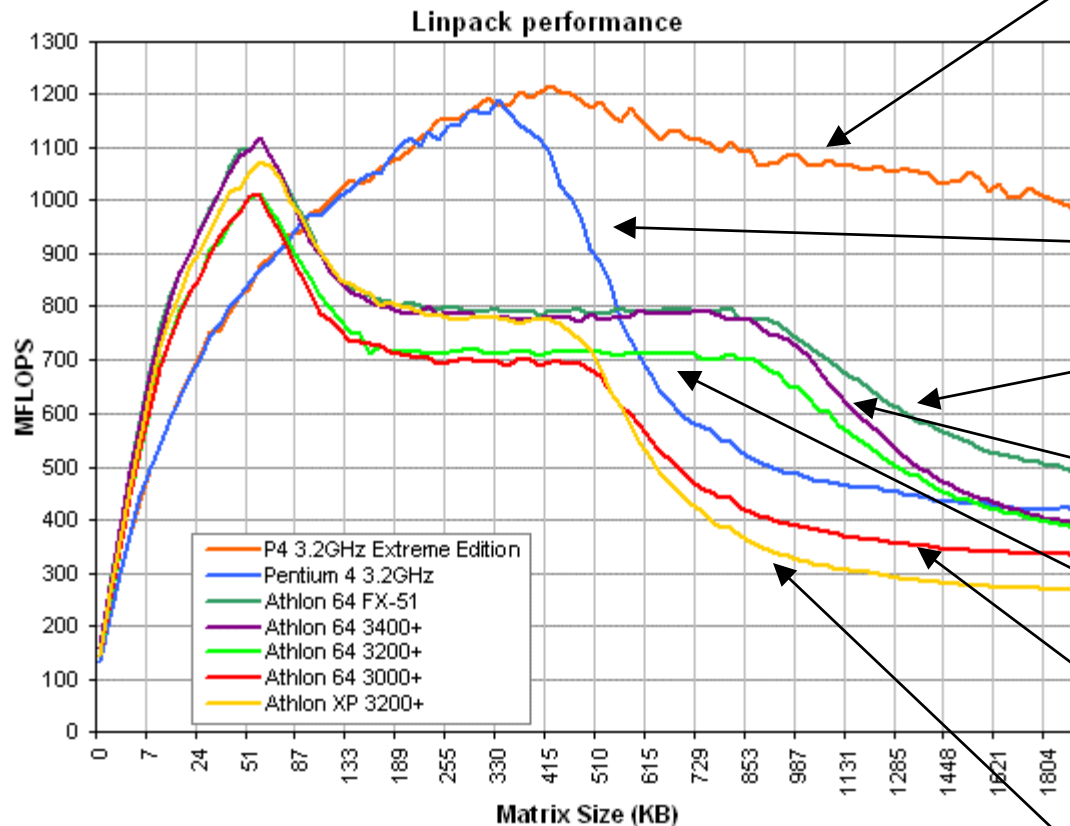
Is memory latency
still an issue?

Source: The Tech Report (1-21-2004)

<http://www.tech-report.com/reviews/2004q1/athlon64-3000/index.x?pg=3>

CMPE550 - Shaaban

X86 CPU Cache/Memory Performance Example: AMD Athlon XP/64/FX Vs. Intel P4/Extreme Edition



Intel P4 3.2 GHz
Extreme Edition
Data L1: 8KB
Data L2: 512 KB
Data L3: 2048 KB

Intel P4 3.2 GHz
Data L1: 8KB
Data L2: 512 KB

AMD Athlon 64 FX51 2.2 GHz
Data L1: 64KB
Data L2: 1024 KB (exclusive)

AMD Athlon 64 3400+ 2.2 GHz
Data L1: 64KB
Data L2: 1024 KB (exclusive)

AMD Athlon 64 3200+ 2.0 GHz
Data L1: 64KB
Data L2: 1024 KB (exclusive)

AMD Athlon 64 3000+ 2.0 GHz
Data L1: 64KB
Data L2: 512 KB (exclusive)

AMD Athlon XP 2.2 GHz
Data L1: 64KB
Data L2: 512 KB (exclusive)

**Main Memory: Dual (64-bit) Channel PC3200 DDR SDRAM
peak bandwidth of 6400 MB/s**

Source: The Tech Report 1-21-2004

<http://www.tech-report.com/reviews/2004q1/athlon64-3000/index.x?pg=3>

CMPE550 - Shaaban