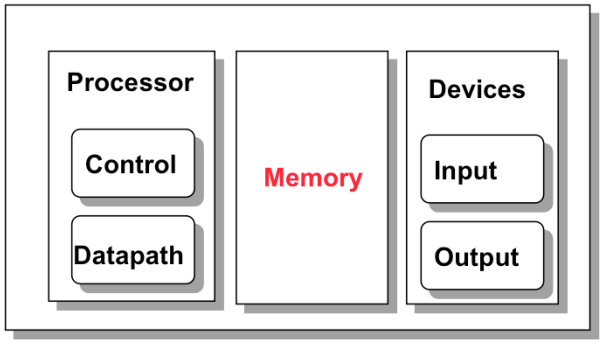
Memory Hierarchy Review

[Adapted from Mary Jane Irwin for
Computer Organization and Design,
Patterson & Hennessy, © 2005, UCB]

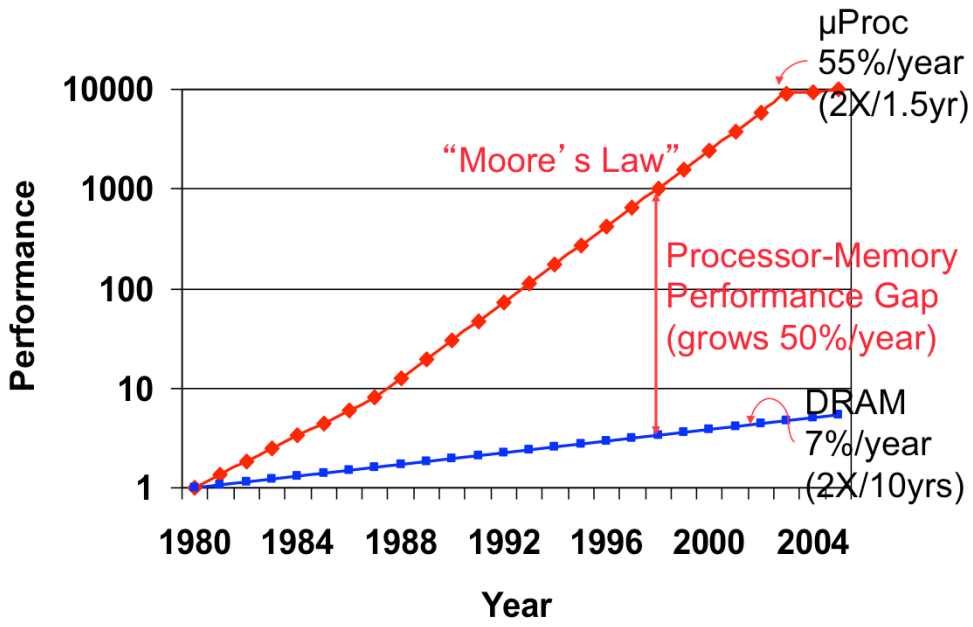
Other handouts
To handout next time

Review: Major Components of a Computer



Workstation Design Target: 25% of cost on Processor, 25% of cost on Memory (minimum memory size), rest on I/O devices, power supplies, box

Processor-Memory Performance Gap



Memory baseline is a 64KB DRAM in 1980, with three years to the next generation until 1996 and then two years thereafter with a 7% per year performance improvement in latency.

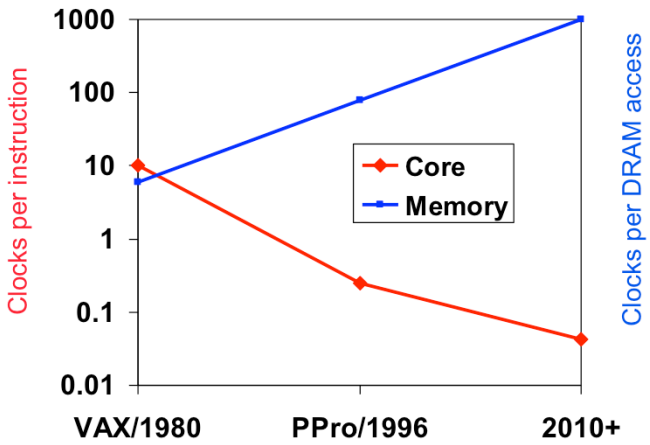
Processor assumes a 35% improvement per year until 1986, then a 55% until 2003, then 5%

Need to supply an instruction and a data every clock cycle

In 1980 there were no caches (and no need for them), by 1995 most systems had 2 level caches (e.g., 60% of the transistors on the Alpha 21164 were in the cache)

The “Memory Wall”

- Logic vs DRAM speed gap continues to grow



The Memory Hierarchy Goal

- Fact: Large memories are slow and fast memories are small

- How do we create a memory that gives the illusion of being large, cheap and fast (most of the time)?
  - With hierarchy
  - With parallelism

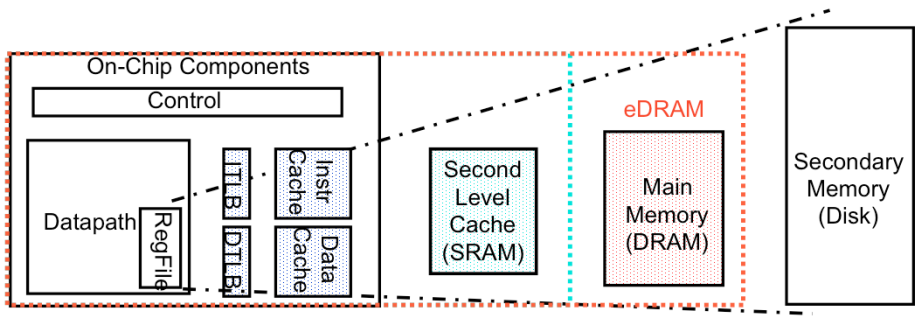
Example of library and selecting books from shelves (spatial correl)
And leaving them on table for fast access
Same book might be used repeatedly (temporal locality)

Concepts of temporal and spatial locality

Done via Memory hierarchy (memories of different sizes and speeds)

A Typical Memory Hierarchy

- By taking advantage of the principle of locality
  - Can present the user with as much memory as is available in the cheapest technology
  - at the speed offered by the fastest technology



Speed (%cycles):	½'s	1's	10's	100's	1,000's
Size (bytes):	100's	K's	10K's	M's	G's to T's
Cost:	highest				lowest

Instead, the memory system of a modern computer consists of a series of black boxes ranging from the fastest to the slowest.

Besides variation in speed, these boxes also varies in size (smallest to biggest) and cost.

What makes this kind of arrangement work is one of the most important principles in computer design: The principle of locality. The principle of locality states that programs access a relatively small portion of the address space at any instant of time.

The design goal is to present the user with as much memory as is available in the cheapest technology (points to the disk).

While by taking advantage of the principle of locality, we like to provide the user an average access speed that is very close to the speed that is offered by the fastest technology.

(We will go over this slide in detail in the next lectures on caches).

caches to improve speed of Virtual address translation

ITLB = Instruction Translation Lookaside Buffer

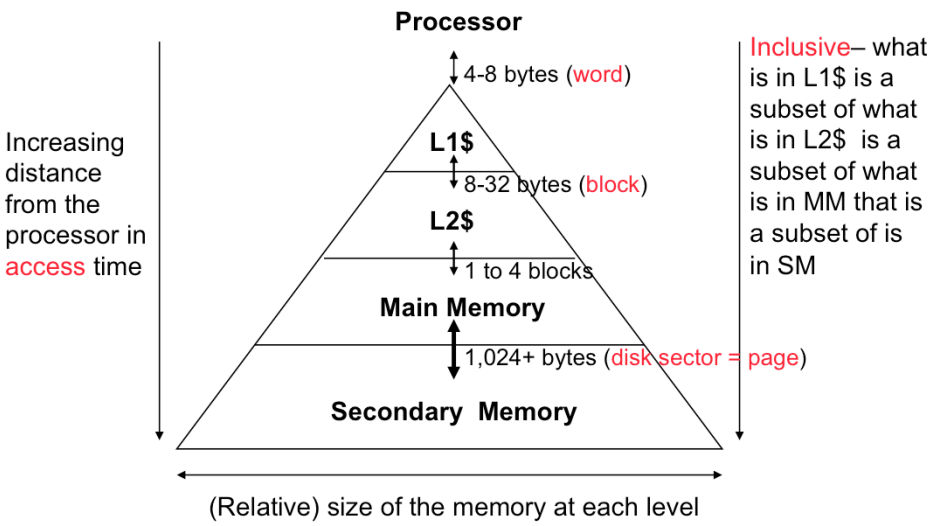
DTLB = Data ….

eDRAM = embedded DRAM (on the same chip as the processor)

DRAM = needs periodic refresh to keep values (high-density transistors)

SRAM = uses latching logic and does not need refresh (low-density transistors)

Characteristics of the Memory Hierarchy



Memory copied from 2 adjacent levels

HIT = when data is found at the upper mem level

MISS = when not found because it is at the lower level

HIT rate is memory hierarchy performance

MISS penalty is the time it takes to copy memory from lower level to upper level + time to deliver block to processor

1,1 ≙ 100%

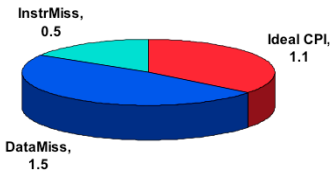
~~1,5~~

0,4

Memory Performance Impact on Performance

- Suppose a processor executes at
  - ideal CPI = 1.1
  - 50% arith/logic, 30% ld/st, 20% control

only ones which use memory



and that 10% of data memory operations miss with a 50 cycle miss penalty

wichtig

- CPI = ideal CPI + average stalls per instruction
  = 1.1(cycle) + (0.30 (datamemops/instr)
  x 0.10 (miss/datamemop) x 50 (cycle/miss))
  = 1.1 cycle + 1.5 cycle = 2.6

  so 58% of the time the processor is stalled waiting for memory!

- A 1% instruction miss rate would add an *additional* 0.5 to the CPI!

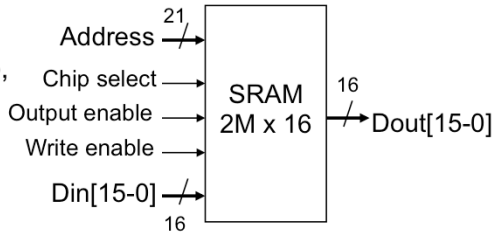
Hit time = time to access the upper level including detecting hit or miss

Miss penalty = time to replace block in the upper level with the block from lower level + time to deliver block to processor

Result: memory hierarchy affects performance

Memory Hierarchy Technologies

- Caches use *SRAM* for speed and technology compatibility
  - Low density (6 transistor cells), high power, expensive, fast
  - Static: content will last "forever" (until power turned off)



- Main Memory uses *DRAM* for size (density)
  - High density (1 transistor cells), low power, cheap, slow
  - Dynamic: needs to be "refreshed" regularly (~ every 8 ms)
    - 1% to 2% of the active cycles of the DRAM
  - Addresses divided into 2 halves (row and column)
    - *RAS* or *Row Access Strobe* triggering row decoder
    - *CAS* or *Column Access Strobe* triggering column selector

Size comparison: DRAM/SRAM is 4 to 8 times

Cost/cycle time comparison SRAM/DRAM is 8 to 16 times

Need output enable on SRAM because outputs are tri-stated (0, 1, high impedance)

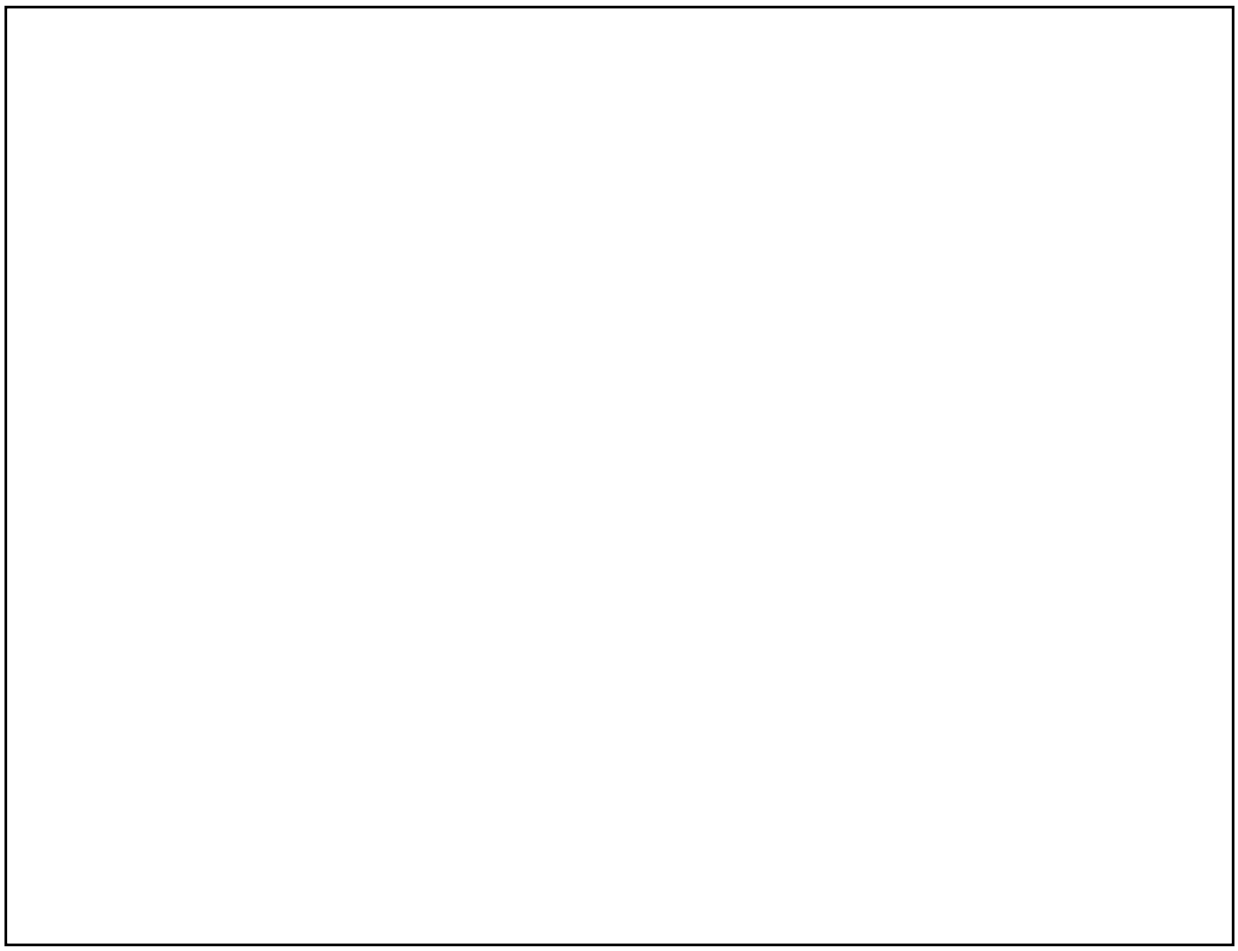
Entire row updated in one cycle

Address lines shared by row and column address

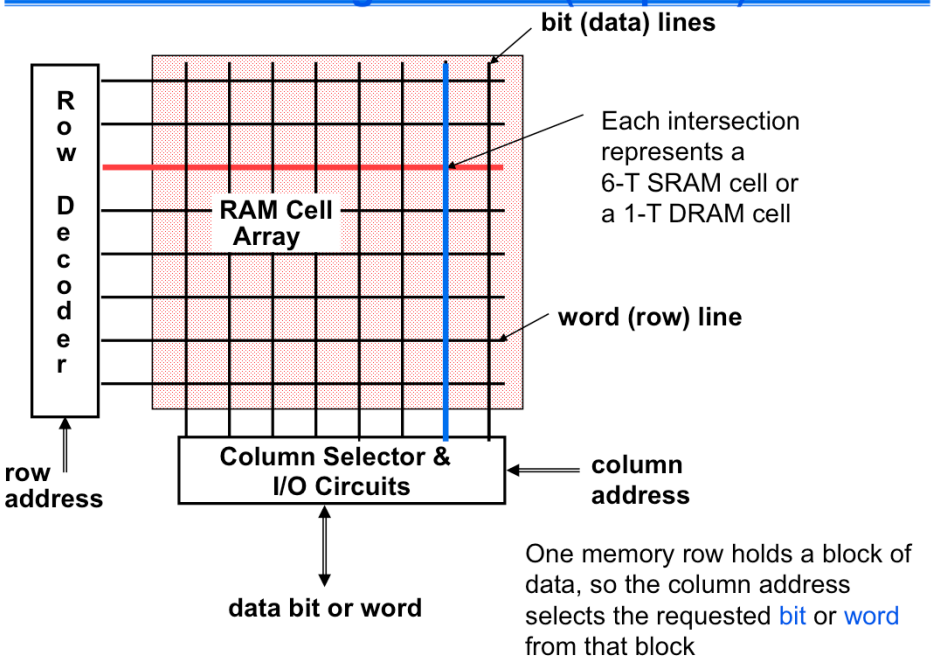
And RAS and CAS provide info on which address it is

Memory Performance Metrics

- Latency: Time to access one word
  - *Access time*: time between the request and when the data is available (or written)
  - *Cycle time*: time between requests
  - Usually cycle time > access time
  - Typical read access times for SRAMs in 2004 are 2 to 4 ns for the fastest parts to 8 to 20ns for the typical largest parts
- Bandwidth: How much data from the memory can be supplied to the processor per unit time
  - width of the data channel * the rate at which it can be used



Classical RAM Organization (~Square)



Put multiple words in one memory row – splits the decoder into two decoders (row and column) and makes the memory core square reducing the length of the bit lines (but increasing the length of the word lines). The lsb part of the address goes into the column decoder (e.g., 6 bits so that 64 words are assigned to one row (with 32 bits per word gives 2**11 bit line pairs) leaving 14 bits for the row decoder (giving 2**14 word lines) for a not quite square array. This scheme is good only for up to 64 Kb to 256 Kb. For bigger memories it is too SLOW because the word and bit lines are too long.

14 bits for ROW 2^14 (2 bits for 1 word and 12 for the address so that it is up to 256Kb)

6 bits for COL (32 bits per word – 5 bits to determine 1 of the 32 bits) = 2^11

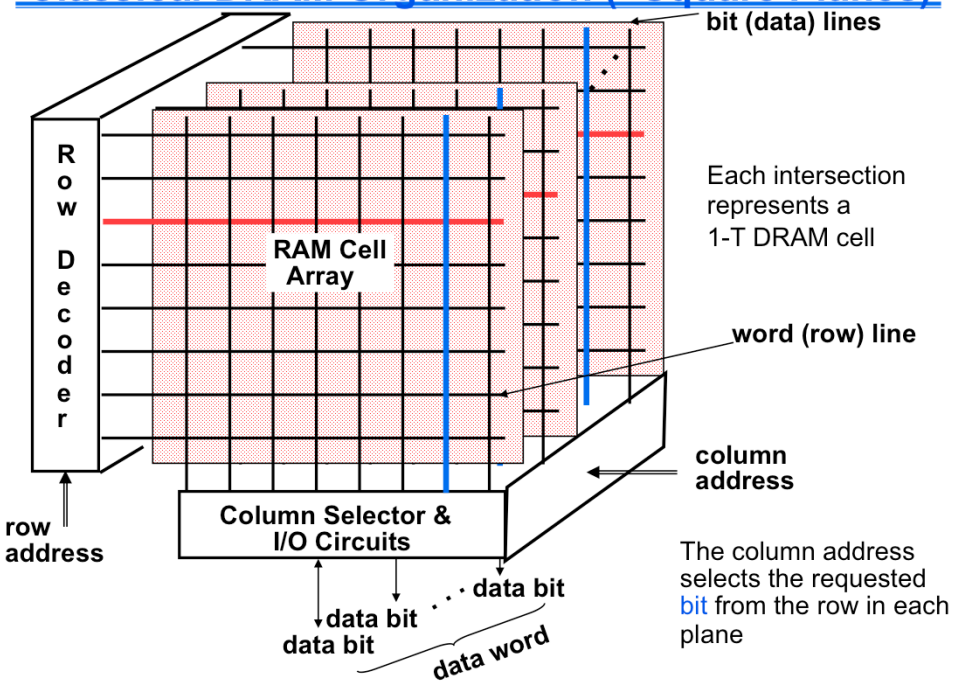
6-T = six transistors

1-T = one transistor

SRAM allows you to read an entire row out at a time at a word.

Each row control line is referred to as the word line and each vertical data line is referred to as the bit line.

Classical DRAM Organization (~Square Planes)



Similar to SRAM, DRAM is organized into rows and columns. But unlike SRAM, which allows you to read an entire row out at a time at a word, classical DRAM only allows you read out one-bit at time.

So we need several (planes) of them to store one word. The reason for this is to save power as well as area.

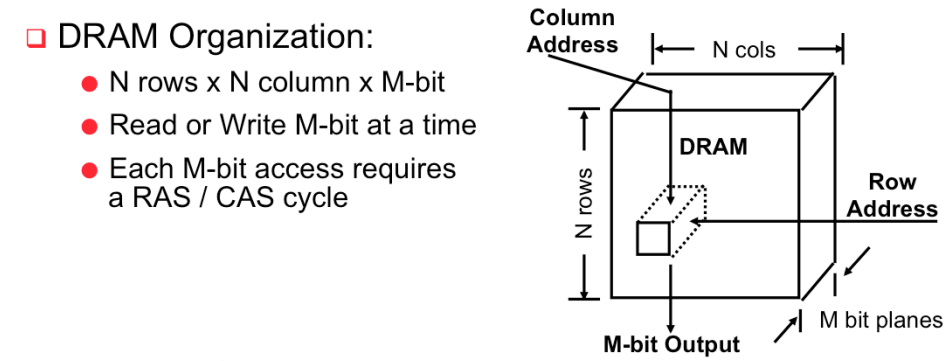
Remember now the DRAM cell is very small we have a lot of them across horizontally.

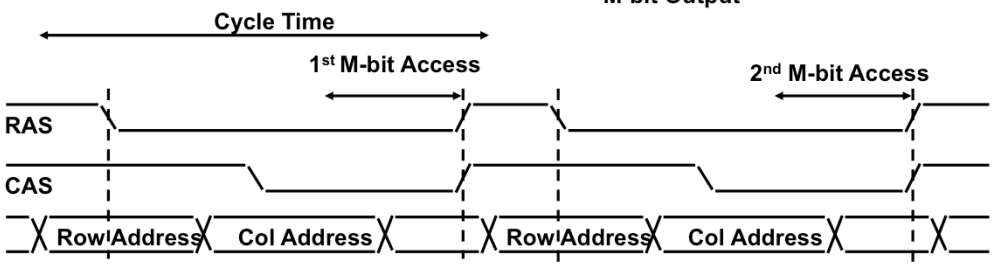
So it will be very difficult to build a Sense Amplifier for each column due to the area constraint not to mention having a sense amplifier per column will consume a lot of power.

You select the bit you want to read or write by supplying a Row and then a Column address. Similar to SRAM, each row control line is referred to as the word line and each vertical data line is referred to as the bit line.

Classical DRAM Operation

- DRAM Organization:
  - N rows x N column x M-bit
  - Read or Write M-bit at a time
  - Each M-bit access requires a RAS / CAS cycle





Another performance booster for DRAM is fast page mode operation.

In normal DRAM, we can only read and write M-bit at time because only one row and one column is selected at any time by the row and column address.

1) RAS

2) Latch

3) Cas

4) Latch

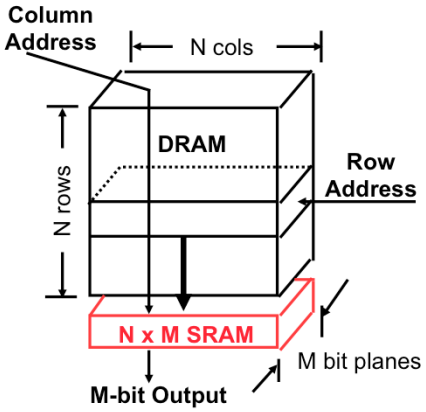
5) Data

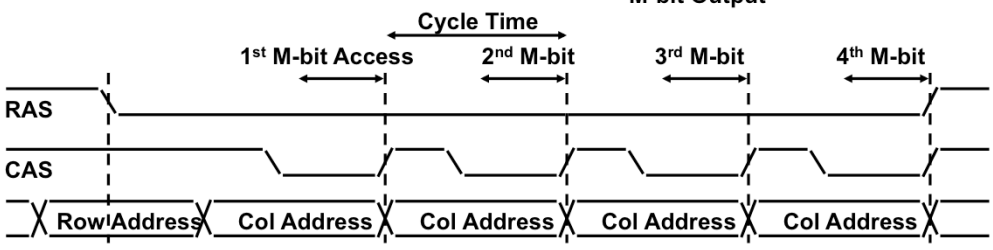
In other words, for each M-bit memory access, we have to provide a row address followed by a column address. Very time consuming.

So the engineers get smart and say: “Wait a minute, this is silly, why don’t we put a N x M register here so we can save an entire row internally whenever we access a row?”

Page Mode DRAM Operation

- Page Mode DRAM
  - N x M SRAM to save a row
- After a row is read into the SRAM "register"
  - Only CAS is needed to access other M-bit words on that row
  - RAS remains asserted while CAS is toggled





So with this register in place, all we need to do is assert the RAS to latch in the row address, then entire row is read out and saved into this register.

After that, you only need to provide the column address and assert the CAS needs to access other M-bit within this same row.

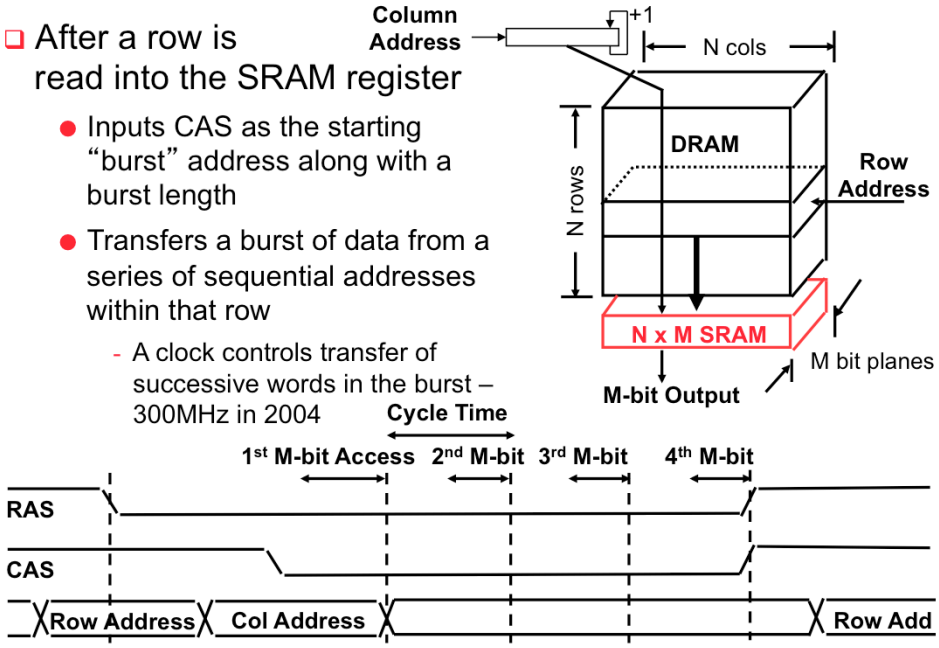
This type of operation where RAS remains asserted while CAS is toggled to bring in a new column address is called Page Mode operation.

Store so don't have to repeat: SRAM

+ 2 = 71 min. (Y:51)

Synchronous DRAM (SDRAM) Operation

- After a row is read into the SRAM register
  - Inputs CAS as the starting "burst" address along with a burst length
  - Transfers a burst of data from a series of sequential addresses within that row
    - A clock controls transfer of successive words in the burst – 300MHz in 2004



Have become the RAM architecture of choice for building memories

Other DRAM Architectures

- Double Data Rate SDRAMs – DDR-SDRAMs (and DDR-SRAMs)
  - Double data rate because they transfer data on both the rising and falling edge of the clock
  - Are the most widely used form of SDRAMs

- DDR2-SDRAMs

DDR2 allows higher bus speed and requires lower power by running the internal clock at half the speed of the data bus

DDR3 is its ability to transfer data at twice the rate (eight times the speed of its internal memory arrays), enabling higher bandwidth or peak data rates

DRAM Memory Latency & Bandwidth Milestones

	DRAM	Page DRAM	FastPage DRAM	FastPage DRAM	Synch DRAM	DDR SDRAM
Module Width	16b	16b	32b	64b	64b	64b
Year	1980	1983	1986	1993	1997	2000
Mb/chip	0.06	0.25	1	16	64	256
Die size (mm^2)	35	45	70	130	170	204
Pins/chip	16	16	18	20	54	66
BWidth (MB/s)	13	40	160	267	640	1600
Latency (nsec)	225	170	125	75	62	52

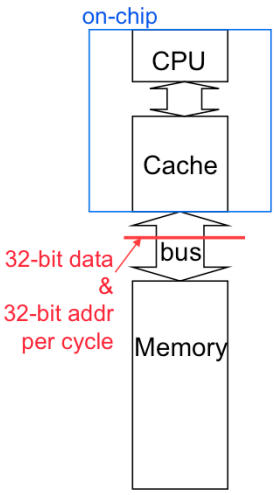
Patterson, CACM Vol 47, #10, 2004

- In the time that the memory to processor bandwidth doubles the memory latency improves by a factor of only 1.2 to 1.4
- To deliver such high bandwidth, the internal DRAM has to be organized as interleaved memory banks

Interleaved banks for parallel access

Memory Systems that Support Caches

- The off-chip interconnect and memory architecture can affect overall system performance in dramatic ways



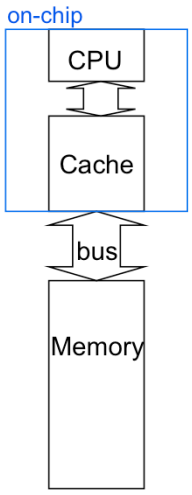
One word wide organization
(one word wide bus and
one word wide memory)

- Assume
  1. 1 clock cycle to send the address
  2. 25 clock cycles for DRAM cycle time, 8 clock cycles access time
  3. 1 clock cycle to return a word of data
- Memory-Bus to Cache bandwidth
  - number of bytes accessed from memory and transferred to cache/CPU per clock cycle

8 clock cycles for the fast page mode access time

Use bandwidth as a performance measure.

One Word Wide Memory Organization



- If the block size is one word, then for a memory access due to a cache miss, the pipeline will have to stall the number of cycles required to return one data word from memory

1	cycle to send address
25	cycles to read DRAM
1	cycle to return data
27	total clock cycles miss penalty

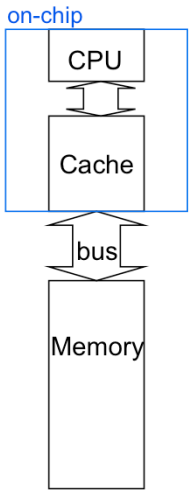
- Number of bytes transferred per clock cycle (bandwidth) for a single miss is

$$4/27 = 0.148 \quad \text{bytes per clock}$$

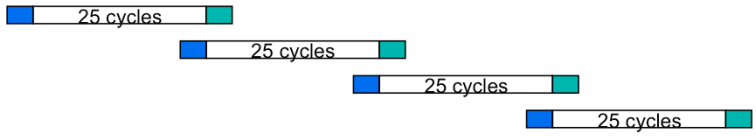
For lecture

One Word Wide Memory Organization, con't

- What if the block size is four words?



	1	cycle to send 1st address
4 x 25 =	100	cycles to read DRAM
	1	cycles to return last data word
	102	total clock cycles miss penalty



- Number of bytes transferred per clock cycle (bandwidth) for a single miss is

  (4 x 4)/102 = 0.157 bytes per clock

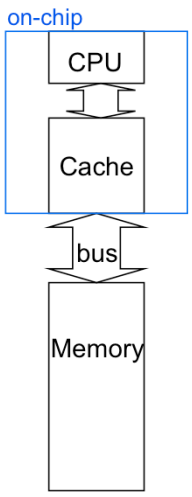
For lecture

Dark blue is the the time to send the address, light blue is the time to return the data word (to processor)

The address can be sent while reading the memory and the data can be transferred to the processor while

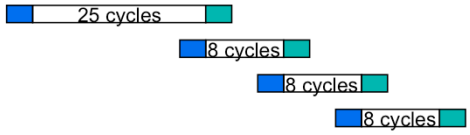
starting to read the memory again.

One Word Wide Memory Organization, con't



- What if the block size is four words and if a fast page mode DRAM is used?

1	cycle to send 1st address
25 + 3*8 = 49	cycles to read DRAM
1	cycles to return last data word
51	total clock cycles miss penalty

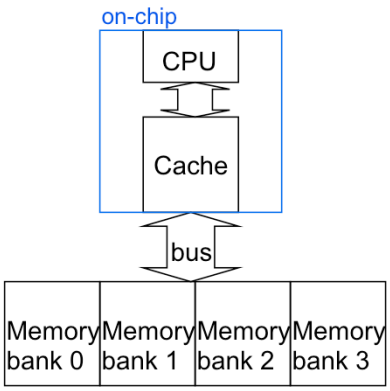


- Number of bytes transferred per clock cycle (bandwidth) for a single miss is
  (4 x 4)/51 = 0.314 bytes per clock

For lecture

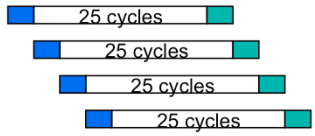
8 cycles for the access time

Interleaved Memory Organization



- For a block size of four words

	1	cycle to send 1st address
25 + 3 =	28	cycles to read DRAM
	1	cycles to return last data word
	30	total clock cycles miss penalty



- Number of bytes transferred per clock cycle (bandwidth) for a single miss is

  (4 x 4)/30 = 0.533 bytes per clock

Bus forces sequence of address data (dark blue) and sequential order of returned data (light blue).

The reading is parallel.

With interleaved memory, memory addresses are allocated to each memory bank in turn. For example, in an interleaved system with 2 memory banks (assuming word-addressable memory) if logical address 32 belongs to bank 0, then logical address 33 would belong to bank 1, logical address 34 would belong to bank 0, and so on. An interleaved memory with n banks is said to be n-way interleaved. If there are n banks, memory location i would reside in bank number i mod n.

Interleaved memory results in contiguous reads (which are common both in multimedia and execution of programs) and contiguous writes (which are used frequently when filling storage or communication buffers) actually using each memory bank in turn, instead of using the same one repeatedly. This results in significantly higher memory throughput as each bank has a minimum waiting time between reads and writes.

DRAM Memory System Summary

- Its important to match the cache characteristics
  - caches access one block at a time (usually more than one word)
- with the DRAM characteristics
  - use DRAMs that support fast multiple word accesses, preferably ones that match the block size of the cache
- with the memory-bus characteristics
  - make sure the memory-bus can support the DRAM access rates and patterns
  - with the goal of increasing the Memory-Bus to Cache bandwidth