

CENG 3420

Computer Organization & Design



Lecture 12: Memory Organization

Bei Yu

CSE Department, CUHK

byu@cse.cuhk.edu.hk

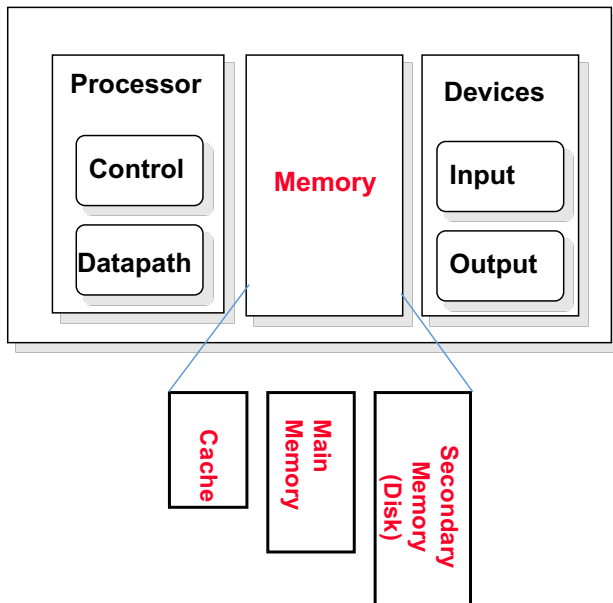
(Textbook: Chapters 5.1–5.2 & A.8–A.9)

Spring 2023



Introduction

Review: Major Components of a Computer





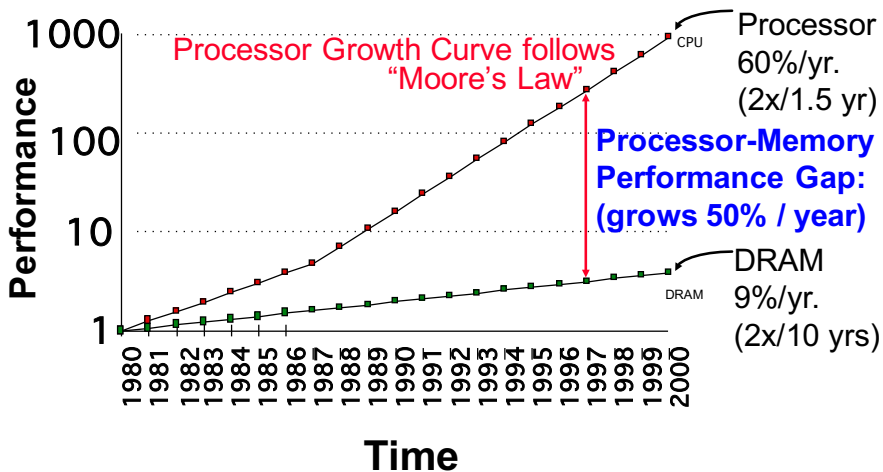
Combinational Circuit:

- Always gives the same output for a given set of inputs
- E.g., adders

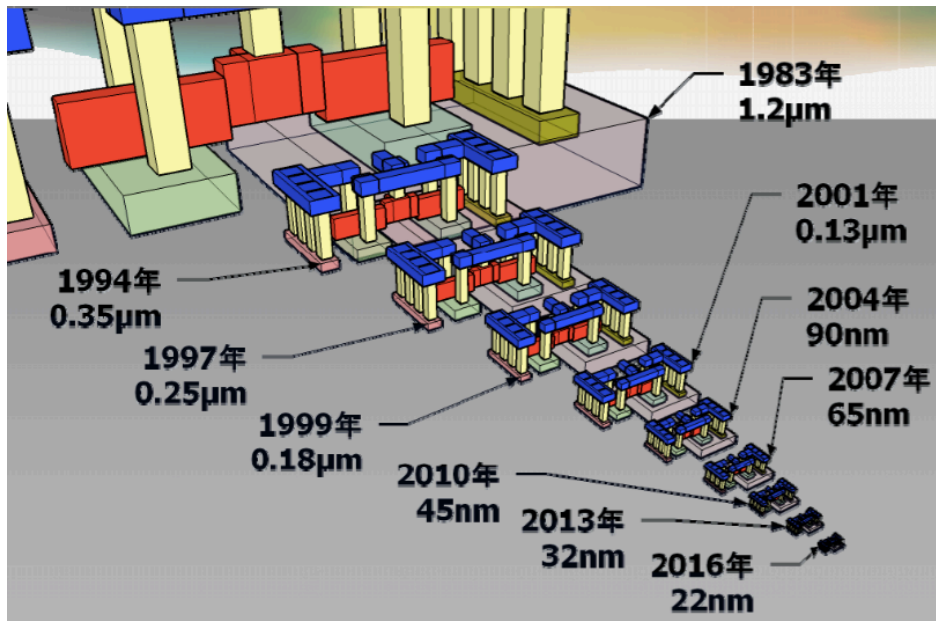
Sequential Circuit:

- Store information
- Output depends on stored information
- E.g., counter
- Need a **storage** element

Who Cares About the Memory Hierarchy?



Processor-DRAM Memory Performance Gap





2005



2014





- Maximum size of memory is determined by addressing scheme

E.g.

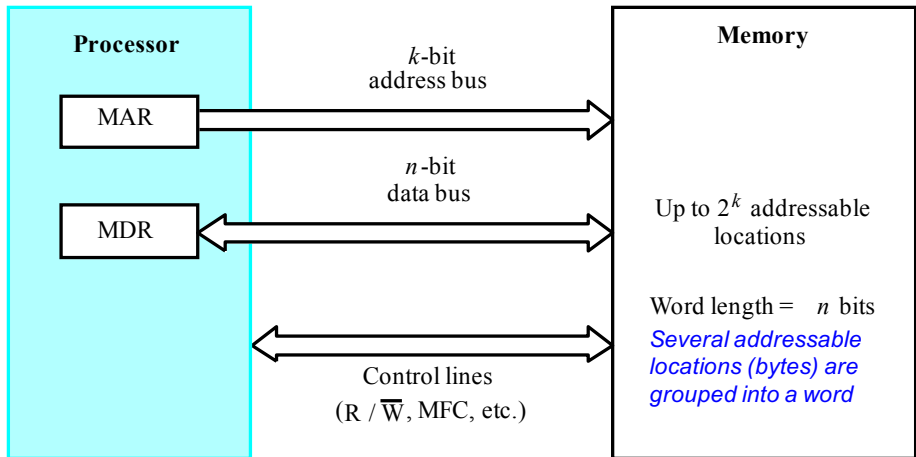
16-bit addresses can only address $2^{16} = 65536$ memory locations

- Most machines are **byte**-addressable
- each memory address location refers to a byte
- Most machines retrieve/store data in words
- Common abbreviations
 - $1k \approx 2^{10}$ (kilo)
 - $1M \approx 2^{20}$ (Mega)
 - $1G \approx 2^{30}$ (Giga)
 - $1T \approx 2^{40}$ (Tera)



Data transfer takes place through

- **MAR**: memory address register
- **MDR**: memory data register





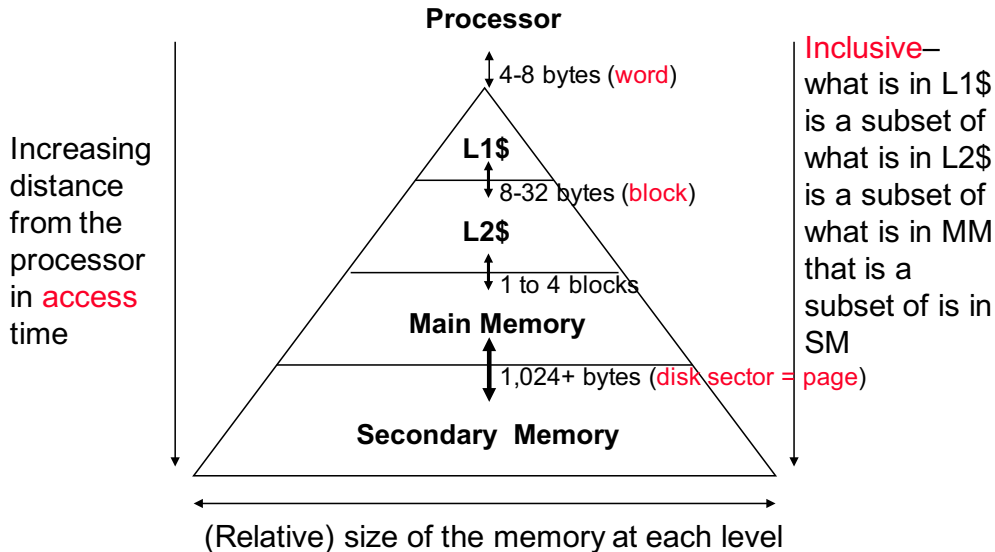
Processor usually runs much faster than main memory:

- Small memories are fast, large memories are slow.
- Use a **cache memory** to store data in the processor that is likely to be used.

Main memory is limited:

- Use **virtual memory** to increase the apparent size of physical memory by moving unused sections of memory to disk (automatically).
- A translation between virtual and physical addresses is done by a memory management unit (**MMU**)
- To be discussed in later lectures

Characteristics of the Memory Hierarchy





Temporal Locality (locality in time)

If a memory location is referenced then it will tend to be referenced again soon

- Keep **most recently accessed** data items closer to the processor



Temporal Locality (locality in time)

If a memory location is referenced then it will tend to be referenced again soon

- Keep **most recently accessed** data items closer to the processor

Spatial Locality (locality in space)

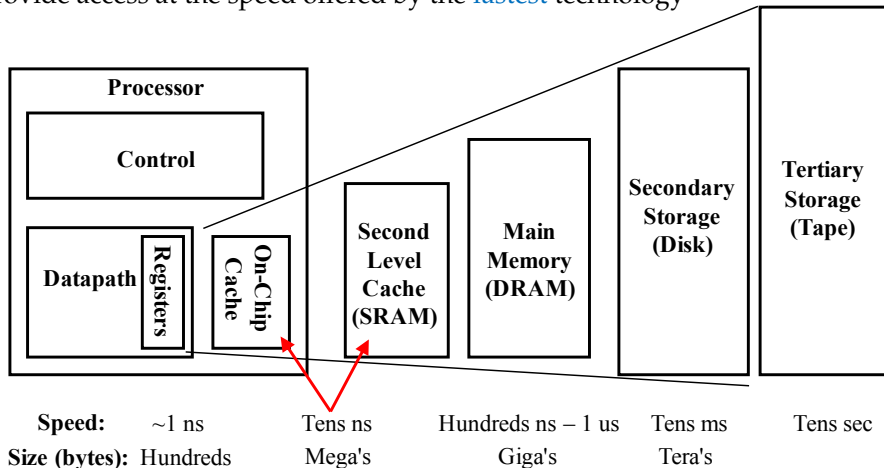
If a memory location is referenced, the locations with nearby addresses will tend to be referenced soon

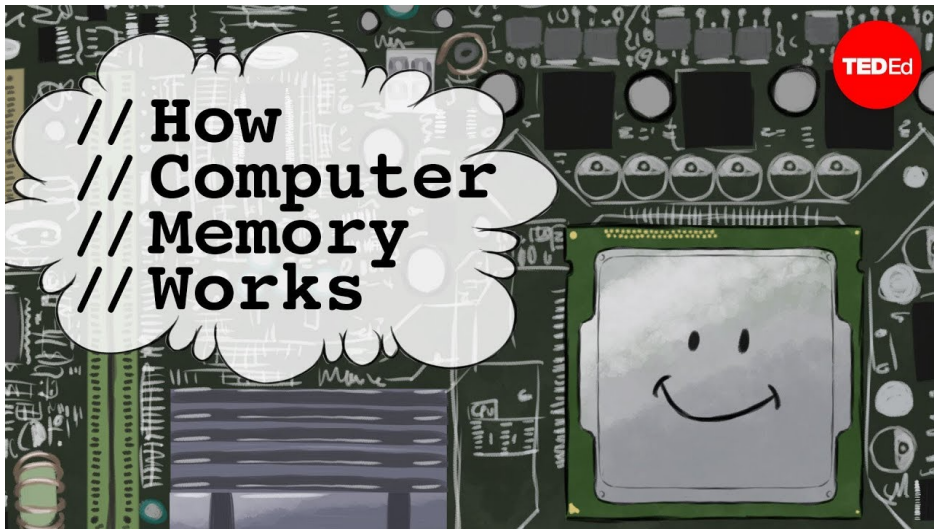
- Move blocks consisting of **contiguous words** closer to the processor



Taking advantage of the **principle of locality**:

- Present the user with as much memory as is available in the **cheapest** technology.
- Provide access at the speed offered by the **fastest** technology





<https://youtu.be/p3q5zWCw8J4>



Random Access Memory (RAM)

Property: comparable access time for any memory locations

Block (or line)

the minimum unit of information that is present (or not) in a cache



- **Hit Rate:** the fraction of memory accesses found in a level of the memory hierarchy
- **Miss Rate:** the fraction of memory accesses not found in a level of the memory hierarchy, i.e. $1 - (\text{Hit Rate})$

Hit Time

Time to access the block + Time to determine hit/miss

Miss Penalty

Time to replace a block in that level with the corresponding block from a lower level

Hit Time \ll **Miss Penalty**



Example

- Mary acts **FAST** but she's always **LATE**.
- Peter is always **PUNCTUAL** but he is **SLOW**.



Example

- Mary acts **FAST** but she's always **LATE**.
- Peter is always **PUNCTUAL** but he is **SLOW**.

Bandwidth:

- talking about the “**number of bits/bytes per second**” when transferring a block of data steadily.

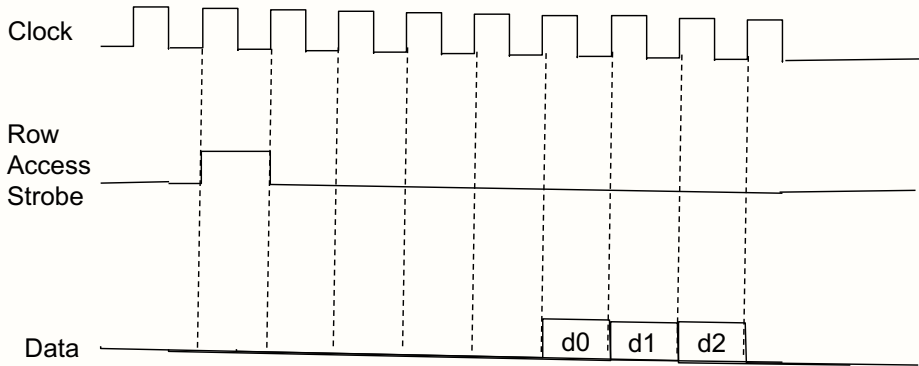
Latency:

- amount of time to transfer the first word of a block after issuing the access signal.
- Usually measure in “**number of clock cycles**” or in $ns/\mu s$.



Question:

Suppose the clock rate is 500 MHz. What is the latency and what is the bandwidth, assuming that each data is 64 bits?

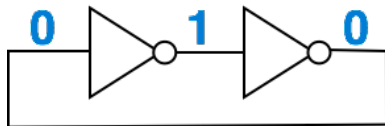




Information Storage

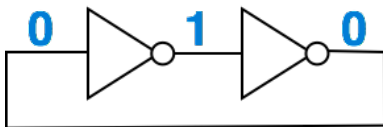


- What if we add feedback to a pair of inverters?

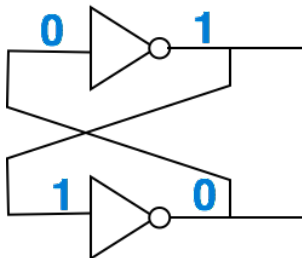




- What if we add feedback to a pair of inverters?



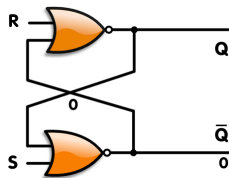
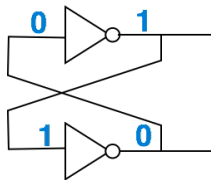
- Usually drawn as a ring of **cross-coupled** inverters
- Stable way to store one bit of information (**w. power**)



How to change the value stored?



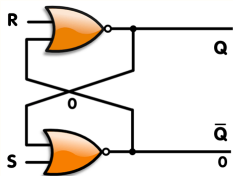
- Replace inverter with **NOR** gate
- **SR-Latch**





QUESTION:

What's the Q value based on different R, S inputs?

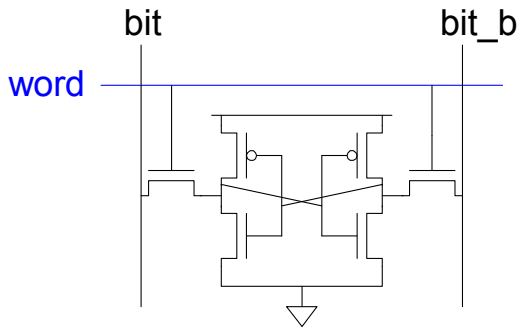


Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

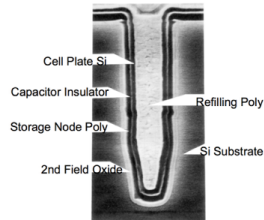
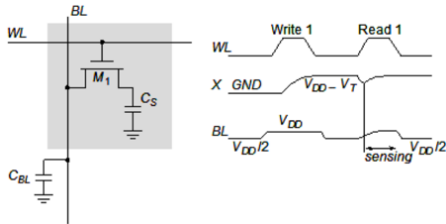
- R=S=1:
- S=0, R=1:
- S=1, R=0:
- R=S=0:



- At least 6 transistors (6T)
- Used in most commercial chips
- A pair of **weak** cross-coupled inverters
- **Data** stored in cross-coupled inverters

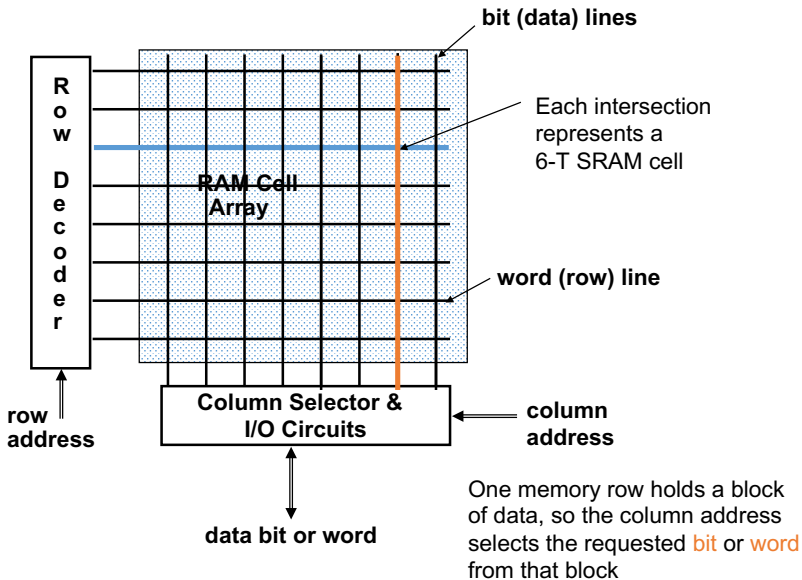


- 1 Transistor (1T)
- Requires presence of an extra capacitor
- Modifications in the manufacturing process.
- Higher density
- **Write:** Charged or discharged the capacitor (slow)
- **Read:** Charge redistribution takes place between bit line and storage capacitance

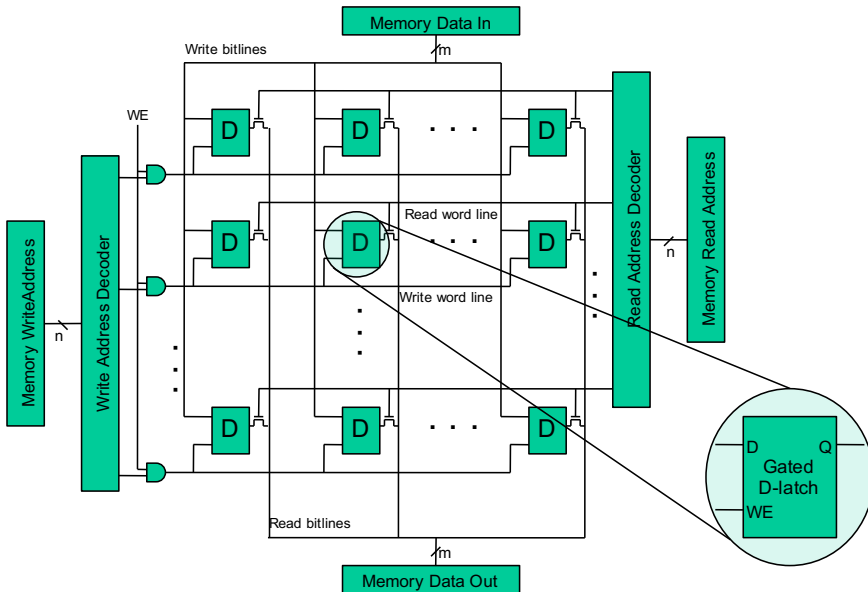




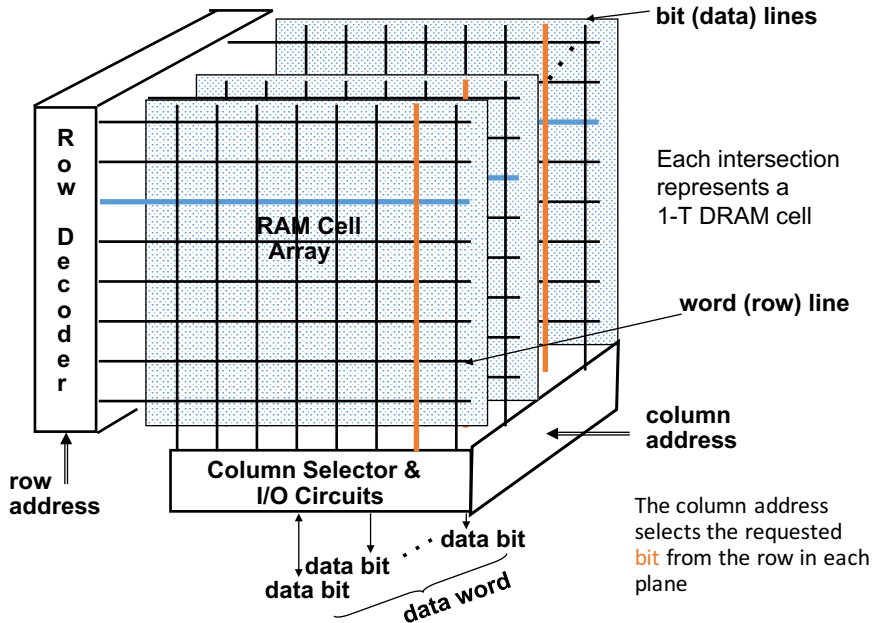
Random Access Memory (RAM)



Latch based memory



Classical DRAM Organization – 3D Structure





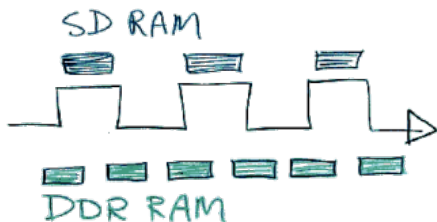
- The common type used today as it uses a clock to synchronize the operation.
- The refresh operation becomes transparent to the users.
- All control signals needed are generated inside the chip.
- The initial commercial SDRAM in the 1990s were designed for clock speed of up to 133MHz.
- Today's SDRAM chips operate with clock speeds exceeding 1 GHz.

Memory modules are used to hold several SDRAM chips and are the standard type used in a computer's motherboard, of size like 4GB or more.





- normal SDRAMs only operate once per clock cycle
- Double Data Rate (DDR) SDRAM transfers data on both clock edges
- **DDR-2** (4x basic memory clock) and **DDR-3** (8x basic memory clock) are in the market.
- They offer increased storage capacity, lower power and faster clock speeds.
- For example, DDR2 can operate at clock frequencies of 400 and 800 MHz. Therefore, they can transfer data at effective clock speed of 800 and 1600 MHz.



1 Hertz

1 Cycle per second

RAM Type	Theoretical Maximum Bandwidth
SDRAM 100 MHz (PC100)	$100 \text{ MHz} \times 64 \text{ bit/cycle} = 800 \text{ MByte/sec}$
SDRAM 133 MHz (PC133)	$133 \text{ MHz} \times 64 \text{ bit/cycle} = 1064 \text{ MByte/sec}$
DDR SDRAM 200 MHz (PC1600)	$2 \times 100 \text{ MHz} \times 64 \text{ bit/cycle} \approx 1600 \text{ MByte/sec}$
DDR SDRAM 266 MHz (PC2100)	$2 \times 133 \text{ MHz} \times 64 \text{ bit/cycle} \approx 2100 \text{ MByte/sec}$
DDR SDRAM 333 MHz (PC2600)	$2 \times 166 \text{ MHz} \times 64 \text{ bit/cycle} \approx 2600 \text{ MByte/sec}$
DDR-2 SDRAM 667 MHz (PC2-5400)	$2 \times 2 \times 166 \text{ MHz} \times 64 \text{ bit/cycle} \approx 5400 \text{ MByte/sec}$
DDR-2 SDRAM 800 MHz (PC2-6400)	$2 \times 2 \times 200 \text{ MHz} \times 64 \text{ bit/cycle} \approx 6400 \text{ MByte/sec}$

Bandwidth comparison. However, due to latencies, SDRAM does not perform as good as the figures shown.



Static RAM (SRAM)

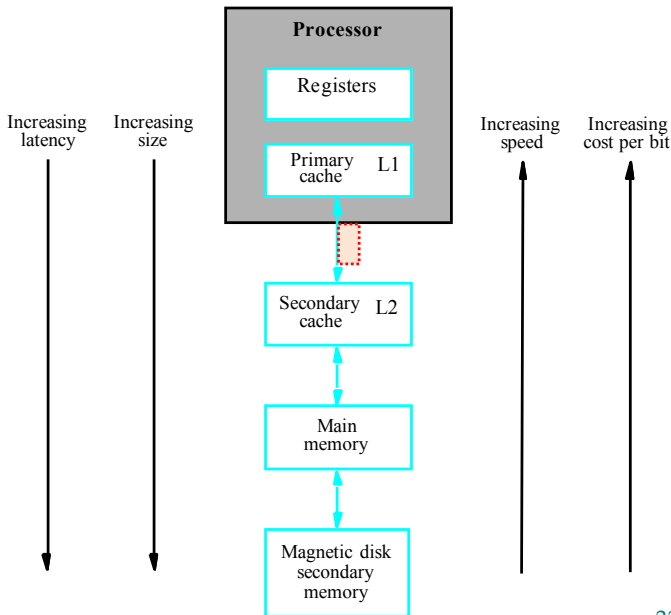
- Capable of retaining the state as long as power is applied.
- They are **fast**, low power (current flows only when accessing the cells) but costly (require several transistors), so the capacity is small.
- They are the Level 1 cache and Level 2 cache inside a processor, of size 3 MB or more.

Dynamic RAM (DRAM)

- store data as electric charge on a capacitor.
- Charge leaks away with time, so DRAMs must be refreshed.
- In return for this trouble, **much higher density** (simpler cells).



- **Aim:** to produce fast, big and cheap memory
- L1, L2 cache are usually SRAM
- Main memory is DRAM
- Relies on *locality of reference*





By taking advantages of the principle of locality:

- Present the user with as much memory as is available in the cheapest technology.
- Provide access at the speed offered by the fastest technology.

DRAM is **slow** but cheap and **dense**:

- Good choice for presenting the user with a BIG memory system – main memory

SRAM is **fast** but expensive and **not very dense**:

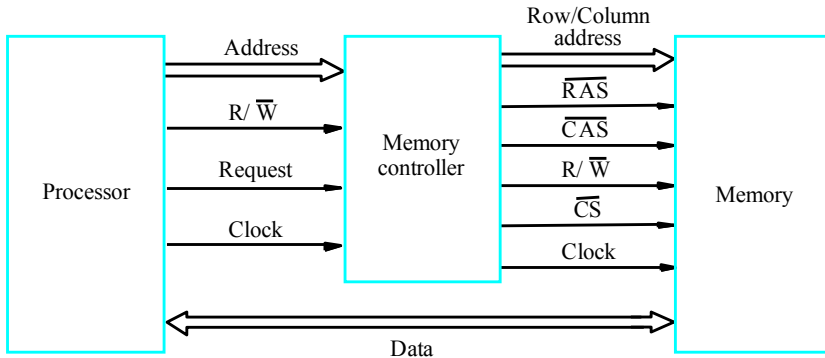
- Good choice for providing the user FAST access time – L1 and L2 cache



Interleaving



- A **memory controller** is normally used to interface between the memory and the processor.
- DRAMs have a slightly more complex interface as they need refreshing and they usually have time-multiplex signals to reduce pin number.
- SRAM interfaces are simpler and may not need a memory controller.



RAS (CAS) = Row (Column) Address Strobe; CS = Chip Select



- The memory controller accepts a complete address and the R/W signal from the processor.
- The controller generates the **RAS** (Row Access Strobe) and **CAS** (Column Access Strobe) signals.



- The memory controller accepts a complete address and the R/W signal from the processor.
- The controller generates the **RAS** (Row Access Strobe) and **CAS** (Column Access Strobe) signals.
- The **high-order** address bits, which select a row in the cell array, are provided first under the control of the RAS (Row Access Strobe) signal.
- Then the **low-order** address bits, which select a column, are provided on the same address pins under the control of the CAS (Column Access Strobe) signal.



- The memory controller accepts a complete address and the R/W signal from the processor.
- The controller generates the **RAS** (Row Access Strobe) and **CAS** (Column Access Strobe) signals.
- The **high-order** address bits, which select a row in the cell array, are provided first under the control of the RAS (Row Access Strobe) signal.
- Then the **low-order** address bits, which select a column, are provided on the same address pins under the control of the CAS (Column Access Strobe) signal.
- The right memory module will be selected based on the address. Data lines are connected directly between the processor and the memory.
- SDRAM needs refresh, but the refresh overhead is only less than 1 percent of the total time available to access the memory.

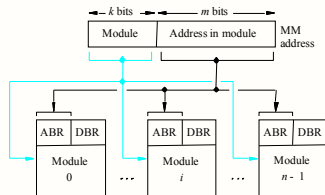


- Processor and cache are fast, main memory is slow.
- Try to hide access latency by **interleaving** memory accesses across several memory modules.
- Each memory module has own Address Buffer Register (**ABR**) and Data Buffer Register (**DBR**)

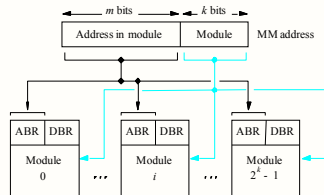


- Processor and cache are fast, main memory is slow.
- Try to hide access latency by **interleaving** memory accesses across several memory modules.
- Each memory module has own Address Buffer Register (**ABR**) and Data Buffer Register (**DBR**)

Which scheme below can be better interleaved?



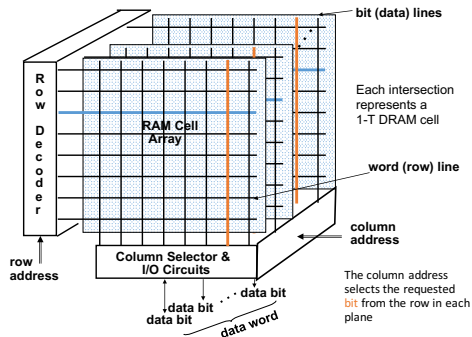
(a) Consecutive words in a module

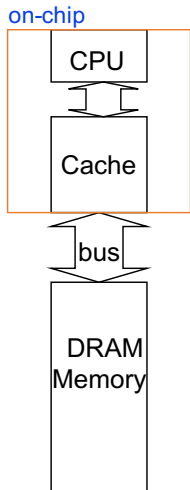


(b) Consecutive words in different modules

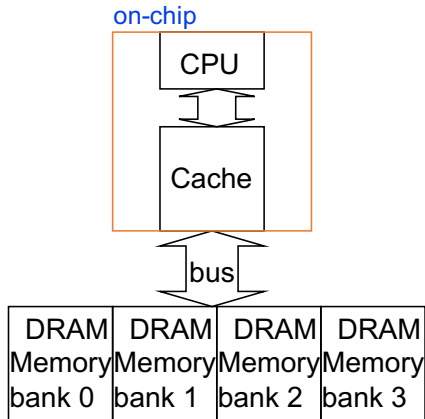


- Two or more **compatible** (identical the best) memory modules are used.
- Within a memory module, several chips are used in “parallel”.
- E.g. 8 modules, and within each module 8 chips are used in “parallel”. Achieve a $8 \times 8 = 64$ -bit memory bus.
- Memory interleaving can be realized in technology such as “**Dual Channel Memory Architecture**”.





(a)
Non-Interleaving



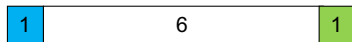
(b) Interleaving



- Suppose we have a cache read miss and need to load from main memory
- Assume cache with 8-word block, i.e., cache line size = 8 words (bytes)
- Assume it takes **one clock** to send address to DRAM memory and **one clock** to send data back.
- In addition, DRAM has **6** cycle latency for first word
- Good that each of subsequent words in same row takes only **4** cycles

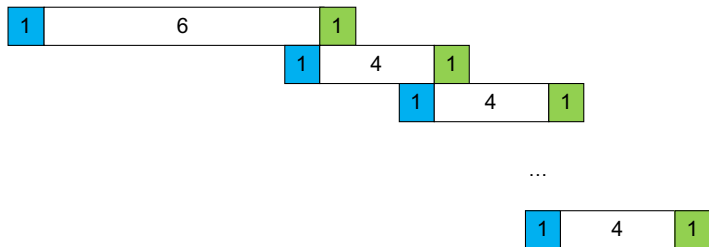
Single Memory Read: $1 + 6 + 1 = 8$ Cycles





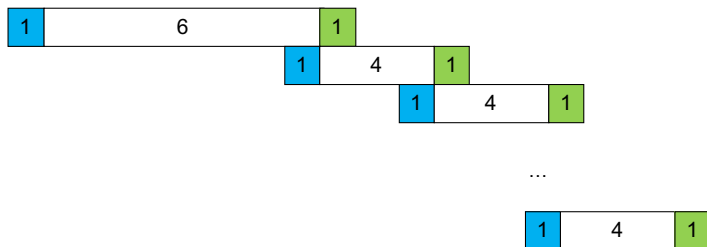
- First byte DRAM needs 6 cycle (same as single memory read)

Example: Non-Interleaving



- First byte DRAM needs 6 cycle (same as single memory read)
- All subsequent words DRAM needs 4 cycle
- Non-overlapping in cache access
- **Assumption:** all words are in the same row

Example: Non-Interleaving

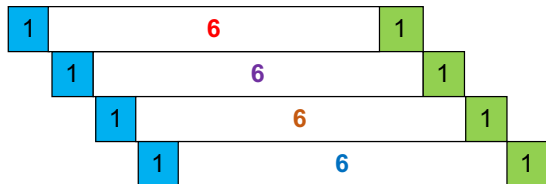


- First byte DRAM needs 6 cycle (same as single memory read)
- All subsequent words DRAM needs 4 cycle
- Non-overlappings in cache access
- **Assumption:** all words are in the same row

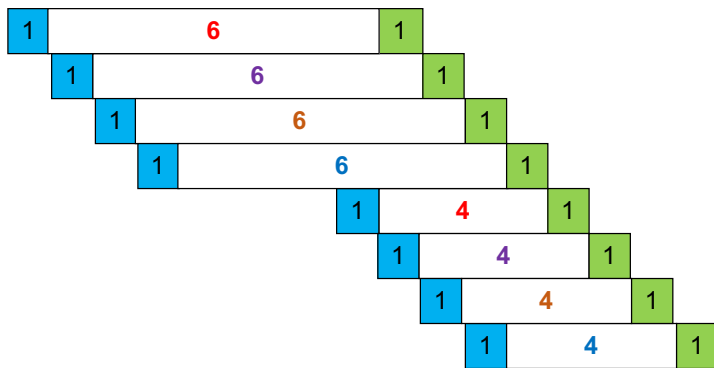
Non-Interleaving Cycle#

$$1 + 1 \times 6 + 7 \times 4 + 1 = 36$$

Example: **Four** Module Interleaving



Example: Four Module Interleaving



Interleaving Cycle#

$$1 + 6 + 1 \times 8 = 15$$



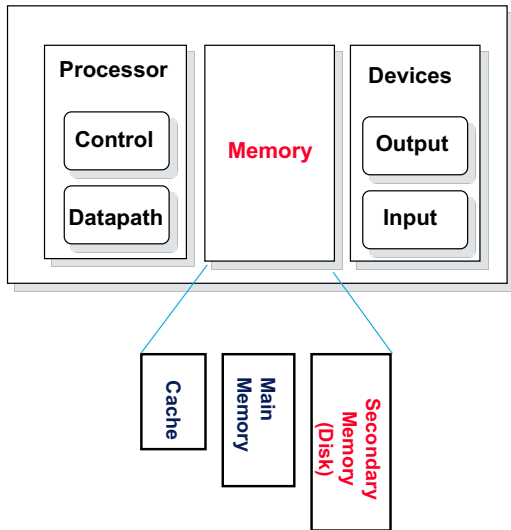
Question:

To transfer 8 bytes, what is the cycle# if just have **TWO**-module interleaved?



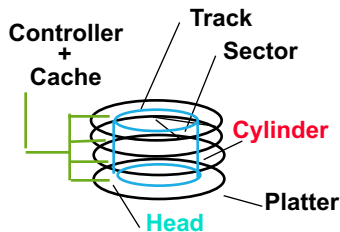
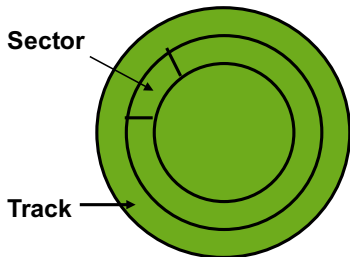
Secondary Memory

Major Components of A Computer



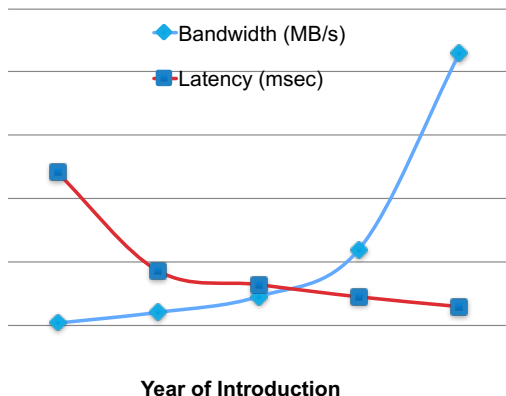


- Long term, **nonvolatile** storage
- Lowest level memory: slow; large; inexpensive
- A rotating platter coated with a magnetic surface
- A moveable read/write head to access the information





- **Latency**: average seek time plus the rotational latency
- **Bandwidth**: peak transfer time of formatted data from the media (not from the cache)



- In the time the bandwidth doubles, latency improves by a factor of only around 1.2



- Memory content fixed and cannot be changed easily.
- Useful to **bootstrap** a computer since RAM is volatile (i.e. lost memory) when power removed.
- We need to store a small program in such a memory, to be used to start the process of loading the OS from a hard disk into the main memory.

PROM/EPROM/EEPROM



- First credible challenger to disks
- Nonvolatile, and $100 \times - 1000 \times$ faster than disks
- **Wear leveling** to overcome **wear out** problem





- Flash devices have greater density, higher capacity and lower cost per bit.
- Can be read and written
- This is normally used for **non-volatile** storage
- Typical applications include cell phones, digital cameras, MP3 players, etc.



- Flash cards are made from FLASH chips
- Flash cards with standard interface are usable in a variety of products.
- Flash cards with USB interface are widely used – memory keys.
- Larger cards may hold 32GB. A minute of music can be stored in about 1MB of memory, hence 32GB can hold 500 hours of music.





Conclusion



- Processor usually runs much faster than main memory
- Common RAM types:
SRAM, DRAM, SDRAM, DDR SDRAM
- Principle of locality: Temporal and Spatial
 - Present the user with as much memory as is available in the **cheapest** technology.
 - Provide access at the speed offered by the **fastest** technology.
- Memory hierarchy:
 - Register → Cache → Main Memory → Disk → Tape