

CENG4480

Lecture 08: Memory 1

Bei Yu

byu@cse.cuhk.edu.hk
(Latest update: November 18, 2019)

Fall 2018



香港中文大學
The Chinese University of Hong Kong

Overview

Introduction

Memory Principle

Random Access Memory (RAM)

Non-Volatile Memory

Conclusion



Overview

Introduction

Memory Principle

Random Access Memory (RAM)

Non-Volatile Memory

Conclusion



Why We Need Memory?

Combinational Circuit:

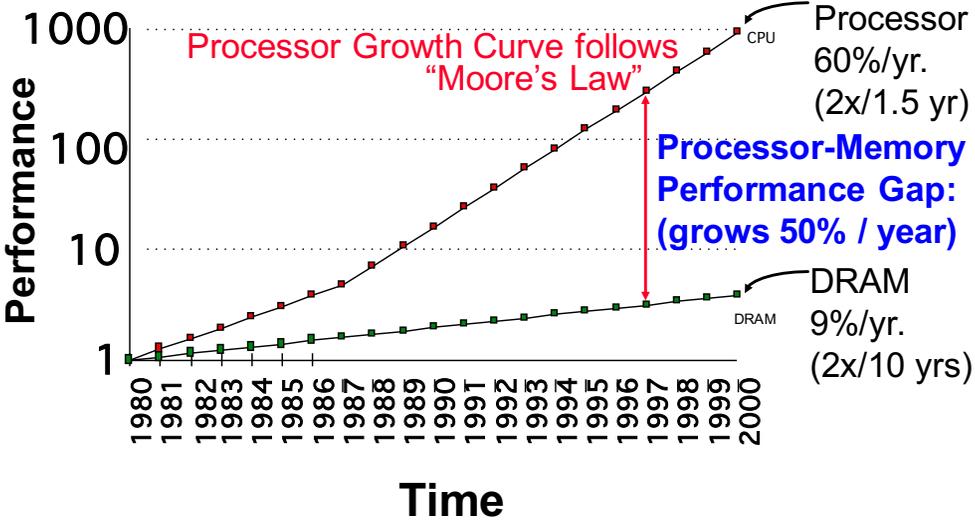
- ▶ Always gives the same output for a given set of inputs
- ▶ E.g., adders

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

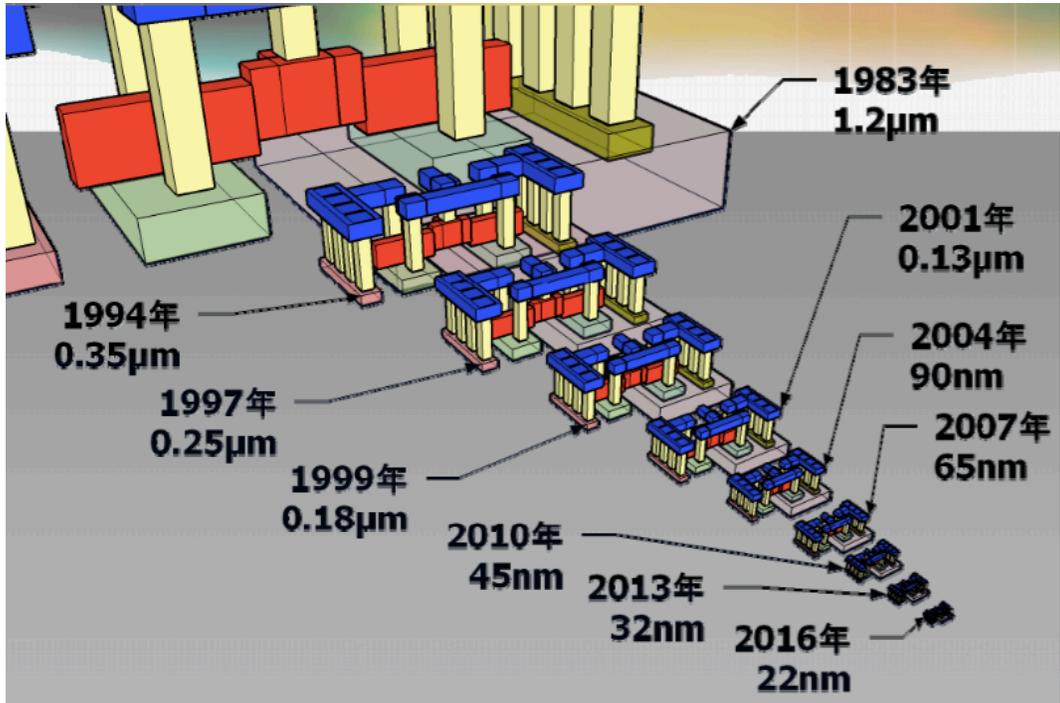


Moore's Law

Transistor number on a unit area would double every 1.5 years.

*1965 paper reprint: [link](#)





2005



2014



Memory System Revisted

- ▶ Maximum size of memory is determined by addressing scheme

E.g.

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

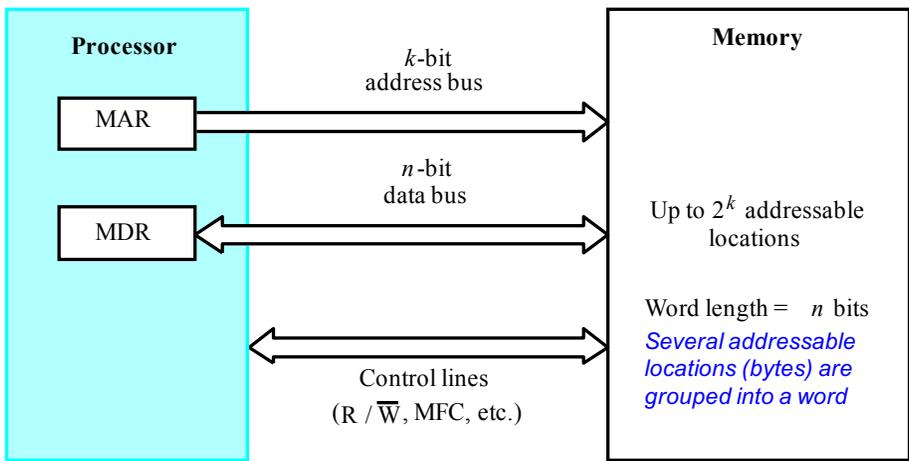
- ▶ Most machines are **byte**-addressable
- ▶ each memory address location refers to a byte
- ▶ Most machines retrieve/store data in words
- ▶ Common abbreviations
 - ▶ $1k \approx 2^{10}$ (kilo)
 - ▶ $1M \approx 2^{20}$ (Mega)
 - ▶ $1G \approx 2^{30}$ (Giga)
 - ▶ $1T \approx 2^{40}$ (Tera)



Simplified View

Data transfer takes place through

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



Big Picture

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:

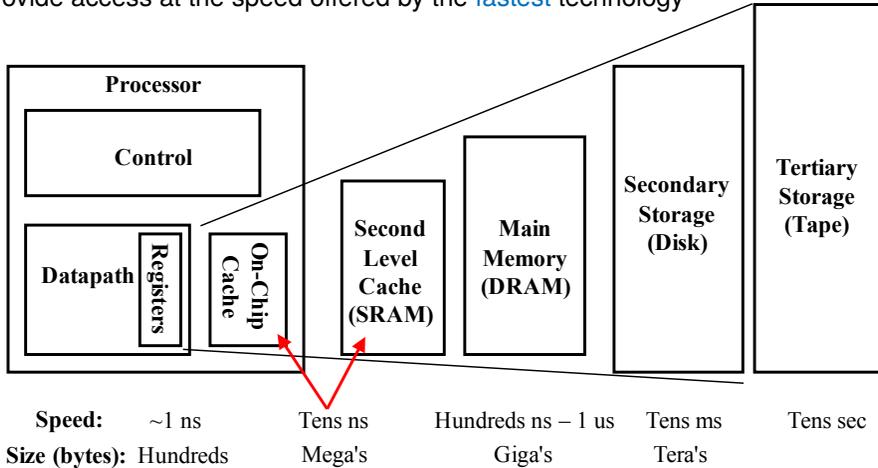
- ▶ Use **virtual memory** to increase the apparent size of physical memory by moving unused sections of memory to disk (automatically).
- ▶ A translation between virtual and physical addresses is done by a memory management unit (**MMU**)
- ▶ To be discussed in later lectures



Memory Hierarchy

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



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 details in the next lecture on caches).

Terminology

Memory Access Time

time between start and finish of a memory request

Memory Cycle Time

minimum delay between successive memory operations

Random Access Memory (RAM)

Property: comparable access time for any memory locations

Overview

Introduction

Memory Principle

Random Access Memory (RAM)

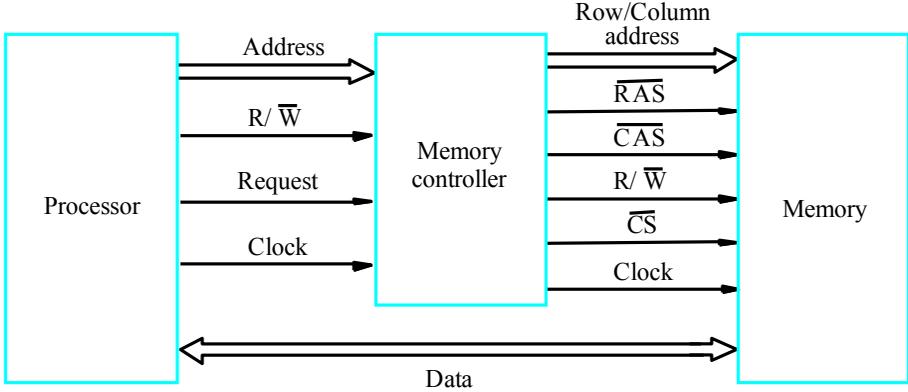
Non-Volatile Memory

Conclusion



Memory Controller

- ▶ 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



Memory Controller

- ▶ 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.

Memory Controller

- ▶ 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.



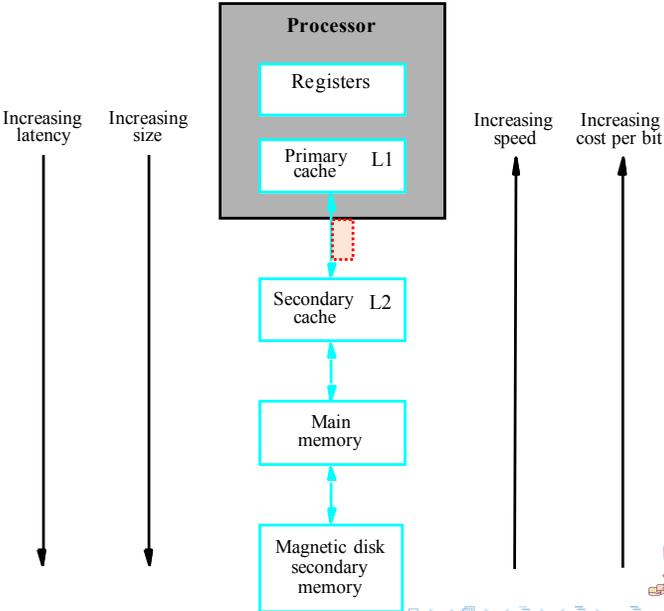
Memory Controller

- ▶ 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.



Memory Hierarchy

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



Locality of Reference

Temporal Locality (locality in time)

- ▶ If an item is referenced, it will tend to be referenced again soon.
- ▶ When information item (instruction or data) is first needed, brought it into cache where it will hopefully be used again.

Spatial Locality (locality in space)

- ▶ If an item is referenced, neighbouring items whose addresses are close-by will tend to be referenced soon.
- ▶ Rather than a single word, fetch data from adjacent addresses as well.



Mix-and-Match: Best of Both

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

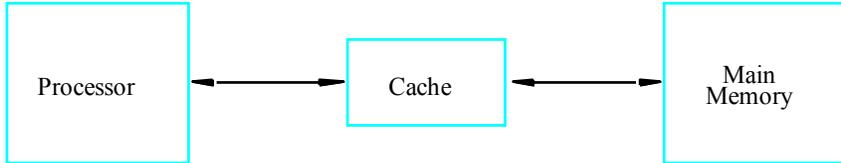
SRAM is fast but expensive and not very dense:

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



Cache Usage

- ▶ Need to determine how the cache is organized
- ▶ **Mapping functions** determine how memory addresses are assigned to cache locations
- ▶ Need to have a **replacement algorithm** to decide what to do when cache is full (i.e. decide which item to be unloaded from cache).



Block

A set of contiguous addresses of a given size (**cache block** is also called **cache line**)



Cache Read Operation

- ▶ Contents of a block are **read** into the cache the first time from the memory.
- ▶ Subsequent accesses are (hopefully) from the cache, called a **cache read hit**.
- ▶ Number of cache entries is relatively small, need to keep most likely used data in cache.
- ▶ When an un-cached block is required, need to employ a **replacement algorithm** to remove an old block and to create space for the new one.



Cache Write Operation

Scheme 1: Write-Through

Cache and main memory updated at the same time.

Note that read misses and read hits can occur.



Cache Write Operation

Scheme 1: Write-Through

Cache and main memory updated at the same time.

Note that read misses and read hits can occur.

Scheme 2: Write-Back

Update cache only and mark the entry dirty. Main memory will be updated later when cache block is removed.

Note that write misses and write hits can occur.



Question 2:

Which write scheme is simpler? Which one has better performance? Why?



Overview

Introduction

Memory Principle

Random Access Memory (RAM)

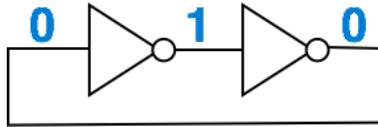
Non-Volatile Memory

Conclusion



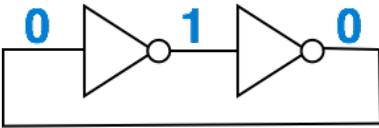
Storage based on Feedback

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

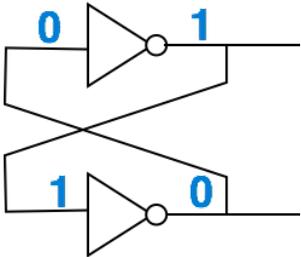


Storage based on Feedback

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



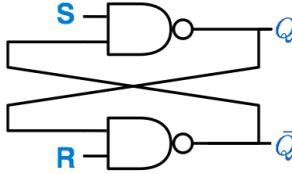
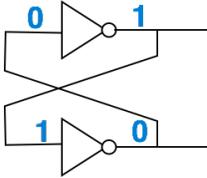
- ▶ Usually drawn as a ring of **cross-coupled** inverters
- ▶ Stable way to store one bit of information (**w. power**)



How to change the value stored?

► Replace inverter with **NAND** gate

► **RS Latch**

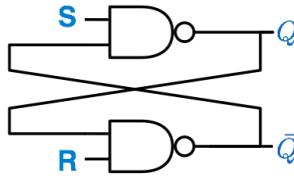


A	B	A nand B
0	0	1
0	1	1
1	0	1
1	1	0



QUESTION:

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



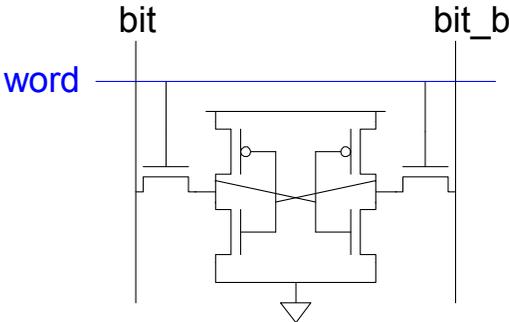
A	B	A nand B
0	0	1
0	1	1
1	0	1
1	1	0

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



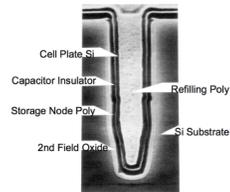
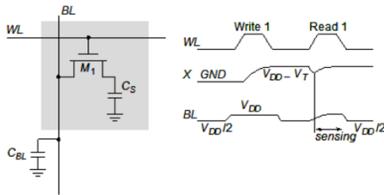
SRAM Cell

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



DRAM Cell

- ▶ 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



SRAM v.s. DRAM

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).



Synchronous DRAM (SDRAM)

- ▶ 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.



Synchronous DRAM (SDRAM)

- ▶ 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.

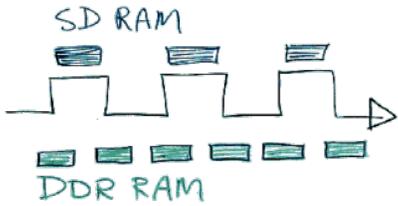


Double Data Rate (DDR) SDRAM

- ▶ 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.



Performance of SDRAM



1 Hertz
1 Cycle per second

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

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



Bandwidth v.s. Latency

Example

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

Bandwidth v.s. Latency

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.

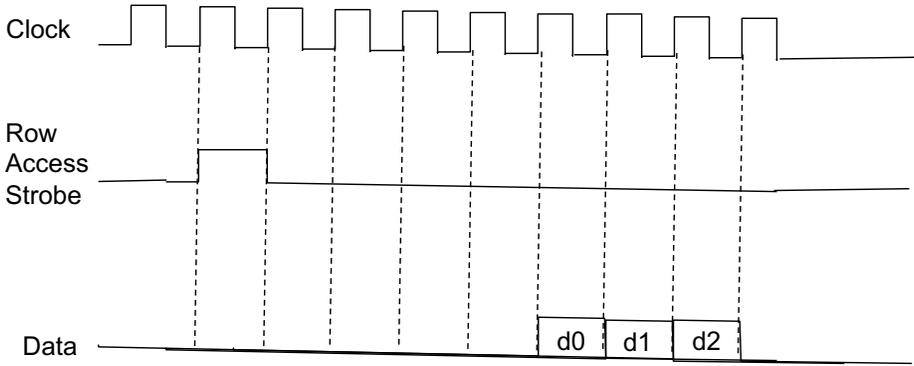
Latency:

- ▶ 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?



$$500\text{MHz} = 2.0 \cdot 10^{-9}\text{s per period}$$

Overview

Introduction

Memory Principle

Random Access Memory (RAM)

Non-Volatile Memory

Conclusion



Read-Only Memory (ROM)

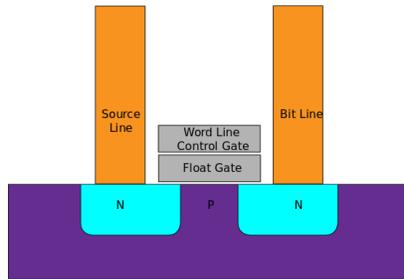
- ▶ Memory content fixed and cannot be changed easily.
- ▶ Useful to **bootstrap** a computer since RAM is volatile (i.e. lost memory) when power removed.
- ▶ We need to store a small program in such a memory, to be used to start the process of loading the OS from a hard disk into the main memory.

PROM/EPROM/EEPROM



FLASH Memory

- ▶ 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.



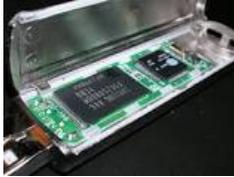
34/38

When the FG is charged with electrons, this charge screens the electric field from the CG, thus, increasing the threshold voltage (V_{T1}) of the cell. This means that now a higher voltage (V_{T2}) must be applied to the CG to make the channel conductive.

The presence of a logical "0" or "1" is sensed by determining whether there is current flowing through the transistor when the intermediate voltage is asserted on the CG.

FLASH Cards

- ▶ 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.



Flash v.s. EEPROM

- ▶ Flash is just one type of EEPROM.
- ▶ Flash uses NAND-type memory, while EEPROM uses NOR type.
- ▶ Flash is block-wise erasable, while EEPROM is byte-wise erasable.
- ▶ Flash is constantly rewritten, while other EEPROMs are seldom rewritten.
- ▶ Flash is used when large amounts are needed, while EEPROM is used when only small amounts are needed.



Arduino Memory

	ATMega168	ATMega328P	ATmega1280	ATmega2560
Flash (1 kByte used for bootloader)	16 kBytes	32 kBytes	128 kBytes	256 kBytes
SRAM	1024 bytes	2048 bytes	8 kBytes	8 kBytes
EEPROM	512 bytes	1024 bytes	4 kBytes	4 kBytes

Overview

Introduction

Memory Principle

Random Access Memory (RAM)

Non-Volatile Memory

Conclusion



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

