CSCI 350
Ch. 12 – Storage Device

Mark Redekopp
Michael Shindler & Ramesh Govindan

Introduction

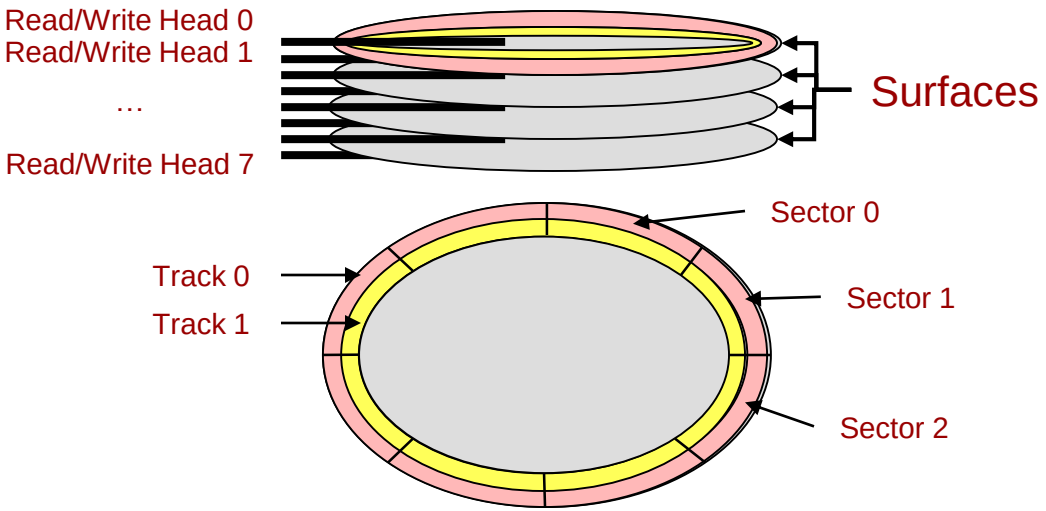
- Storage HW limitations
  - Poor random-access
  - Asymmetric read/write performance
  - Reliability issues
- File system designers and application writers need to understand the hardware

MAGNETIC DISK STORAGE

Magnetic Disk Organization

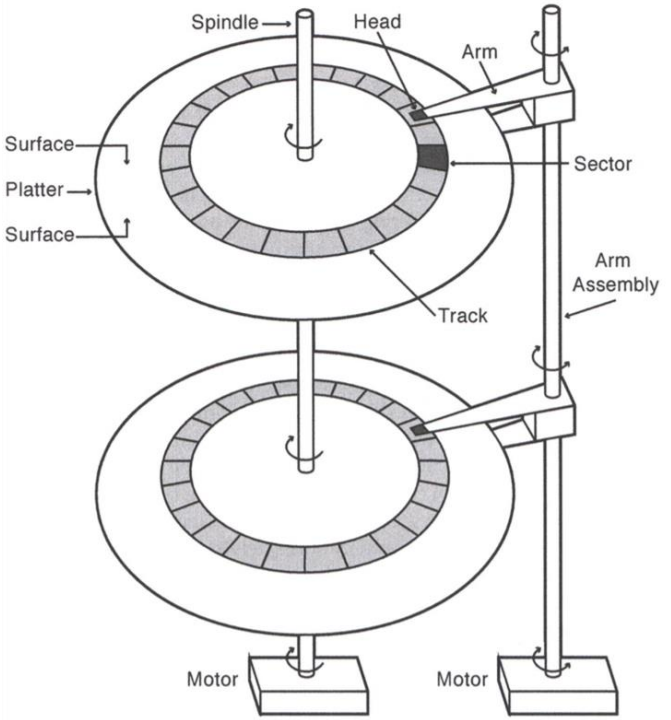
- Double sided surfaces/platters
  - Magnetic coding on metallic film mounted on ceramic/aluminum surface
- Each platter is divided into concentric **tracks** of small **sectors** that each store several thousand bits
- Platters are spun (4,500-15,000 RPMs = 70-250 RPS) and allow the read/write head to skim over the surface inducing a magnetic field and reading/writing bits
- Reading/writing occurs at granularity of ENTIRE sector [usually 512 bytes] not individual bytes

• **Seek Time: Time needed to position the read-head above the proper track**	**3-12 ms**
• **Rotational delay: Time needed to bring the right sector under the read-head**	**5-6 ms**
◦ **Depends on rotation speed (e.g. 5400 RPM)**	
• **Transfer Time:**	**0.1 ms**
• **Disk Controller Overhead:**	**+ 2.0 ms**
	~20 ms

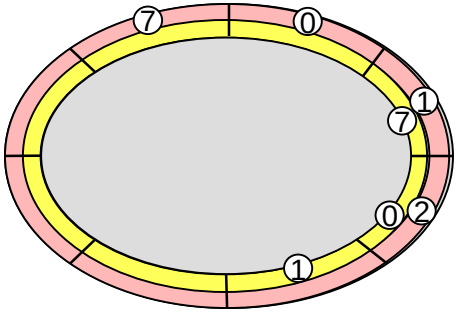


Improving Performance

- Track Skewing
  - Offset sector 0 on neighboring track to allow fast sequential read accounting for the time it takes to move the read head to the next track
- On-board RAM to act as cache
  - Track buffer:
    - When head arrives at desired track it may not be at the right sector
    - Still start reading immediate & store the entire track in on-board memory in case they are wanted later without reading them at that point
  - Write Acceleration
    - Store write data in a cache and return to OS, performing the writes at a more convenient time (Can lead to data loss if power-loss or crash)
    - Tag Command Queueing: OS batches writes and communicates the entire batch to the disk which can re-order them as desired to be optimally scheduled



OS:PP 2nd Ed. Fig. 12.2



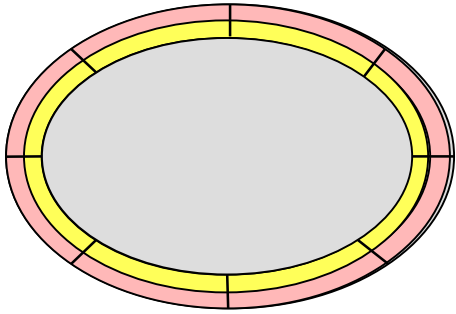
Track skewing: Sector 0 is offset on subsequent tracks based on the rotation speed and time it takes to move the head to the next track

Disk Access Times

- Access time = Seek + Rotation + Transfer Time
- Seek time
  - Time to move head to correct track
    - Mechanical concerns: Include time to wait for arm to stop vibrating and then make finer grained adjustments to position itself correctly over the track
  - Seek time depends on how far the arm has to move
  - Min. seek time approx. 0.3-1.5ms
  - Max. seek time approx. 10-20ms
  - Average seek time (time to move 1/3 of the way across the disk)
- Head transition time
  - If reading track t on one head (surface) and we want to read track t on another do we have to move the arm?

Disk Access Times (Cont.)

- Access time = Seek + Rotation + Transfer Time
- Rotation time
  - Time to rotate the desired starting sector under the head
  - For 4,200 to 15,000 RPM it takes 7.5-2ms for a half rotation of the surface (a reasonable estimation for rotation time)
  - Can use track buffering
- Transfer time
  - Time for the head to read one sector (FAST = few microseconds) into the disks RAM
  - Since outer tracks have more sectors (yet constant rotation speed), outer track bandwidth is higher than inner track
  - Then we must transfer from the disk's RAM to the processor over the computer system's memory
    - Depends on I/O bus (USB 2.0 = 60MB/s, SATA3 = 600 MB/s)*



*Src: https://en.wikipedia.org/wiki/List_of_device_bit_rates#Storage

Example: Random Reads

- Time for 500 random sector reads in FIFO order (no re-scheduling)
  - Seek: Since random locations, use average seek time of 10.5 ms
  - Rotation: At 7200 RPM, 1 rotation = 8.3 ms; Use half of that value 4.15 for average rotation time
  - Transfer: 512 bytes @ 54MB/s = 9.5 us
  - Time per req.: 10.5 + 4.15 + 0.0095 ms = 14.66ms
  - **Total time = 14.66 * 500 = 7.33s**

Size	
Platters/Heads	2/4
Capacity	320 GB
Performance	
Spindle speed	7200 RPM
Average seek time read/write	10.5 ms / 12.0 ms
Maximum seek time	19 ms
Track-to-track seek time	1 ms
Transfer rate (surface to buffer)	54–128 MB/s
Transfer rate (buffer to host)	375 MB/s
Buffer memory	16 MB
Power	
Typical	16.35 W
Idle	11.68 W

OS:PP 2nd Ed. Fig. 12.3

Laptop HD specs.
(Toshiba MK3254GSY)

Example: Sequential Reads

- Time for 500 sequential sector reads (assume same track)
  - Seek: Since we don't know the track, use average seek time of 10.5 ms
  - Rotation: At 7200 RPM, 1 rotation = 8.3 ms; Use half of that value 4.15 for average rotation time since we don't know where the head will be in relation to the desired start sector
  - Transfer:
    - 500 sectors * 512 bytes/sector * 1s/54MB = 4.8 ms
    - 500 sectors * 512 bytes/sector * 1s/128MB = 2 ms
  - **Total time (54MB/s) = 10.5+4.15+4.8=19.5 ms**
  - **Total time (128MB/s) = 10.5+4.15+4.8=16.7 ms**
  - **Actually slightly better due to track buffering**

Size	
Platters/Heads	2/4
Capacity	320 GB
Performance	
Spindle speed	7200 RPM
Average seek time read/write	10.5 ms / 12.0 ms
Maximum seek time	19 ms
Track-to-track seek time	1 ms
Transfer rate (surface to buffer)	54–128 MB/s
Transfer rate (buffer to host)	375 MB/s
Buffer memory	16 MB
Power	
Typical	16.35 W
Idle	11.68 W

OS:PP 2nd Ed. Fig. 12.3

Laptop HD specs. (Toshiba MK3254GSY)

- **Using the 16.7 ms total time we are achieving => 15.33 MB/s**
- **But max rate is 54-128 MB/s**
- **We are achieving a small fraction of max BW**

Disk Scheduling

- FIFO
  - Can yield poor performance for consecutive requests on disparate tracks
- SSTF/SPTF (Shortest Positioning/Seek Time First)
  - Go to the request that we can get to the fastest (like Shortest Job First)
  - Problem 1: Can lead to starvation
  - Problem 2: Unlike SJF it is not optimal
    - Example: Read n sectors that are distance D away in one direction and 2*n sectors at D+1 distance in the opposite direction
    - For response time per request it would be better to first handle the 2n sectors that are d+1 distance then the n sectors but SSTF/SPTF would choose the n sectors first

Disk Scheduling – Elevator Algorithms 1

- Elevator algorithms
- SCAN/CSCAN: Elevator-base algorithms
  - SCAN: Service all requests in the order encountered as the arm moves from inner to outer tracks and then back again (i.e. scan in both forward and reverse directions)
  - CSCAN: Same as SCAN but when we reach the end we return to starting position (w/o servicing requests) and start SCAN again (i.e. only SCAN 1 way)
    - Likely few requests on the end we just serviced (more pending requests back at the start)
    - More fair

Disk Scheduling – Elevator Algorithms 2

- RSCAN/RCSCAN: Rotationally-aware SCAN or CSCAN
- Allows for slight diversions from strict SCAN order based on rotation distance to a sector
- Example: Assume head location on track 0, sector 0
  - Request 1: Track 0, Sector 1000
  - Request 2: Track 1, Sector 500
  - Request 3: Track 10, Sector 0
  - RSCAN/RCSCAN would allow a servicing order of 2, 1, 3 rather than 1,2,3 according to strict SCAN

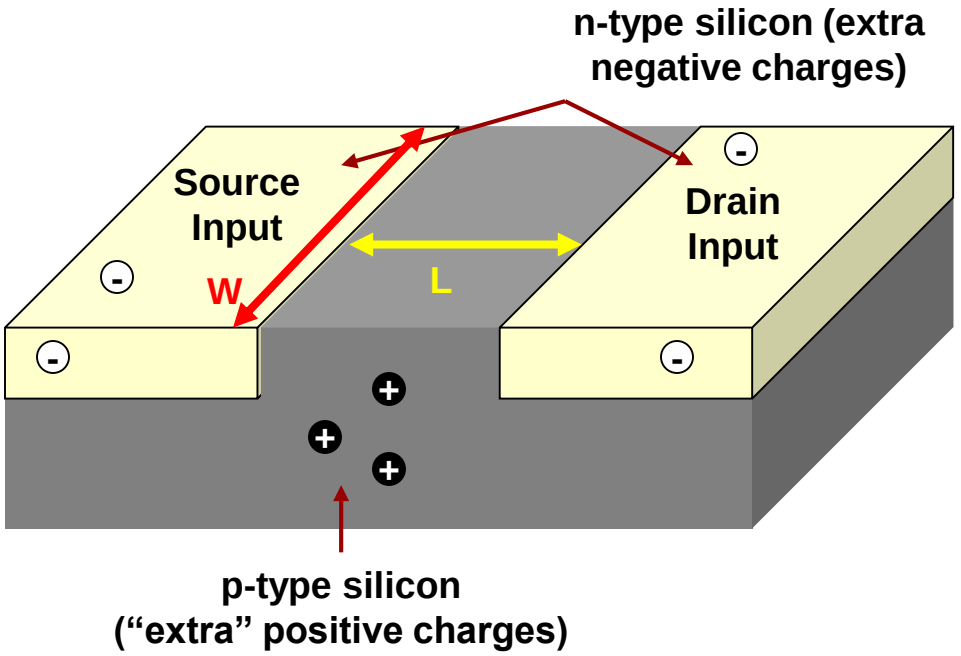
Effect of Disk Scheduling

- Recall time for 500 random sector reads was around 7.3 seconds
- Recalculate using SCAN
  - Seek: Now each seek will be 0.2% of the time to seek across disk. We can interpolate between the minimum track seek (moving over 1 track) and the average 33.3% seek time. This yields 1.06ms
  - Rotation time: Still half the rotation time = 4.15ms
  - Transfer time: Still .0095 ms
  - Time per request = 1.06+4.15+.0095 = 5.22ms
  - Total time = 500*5.22ms = 2.6 seconds
  - Speedup of around 3x for SCAN

FLASH STORAGE

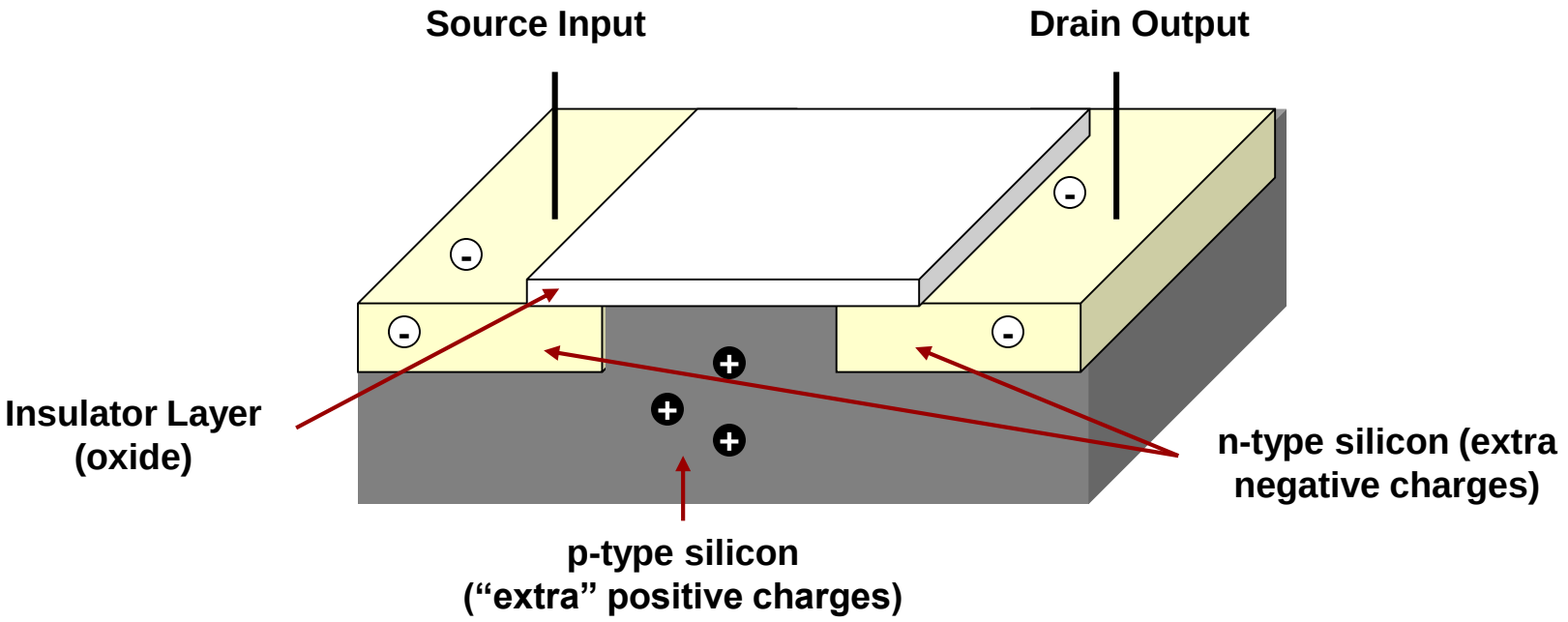
Transistor Physics

- Transistor is started by implanting two n-type silicon areas, separated by p-type



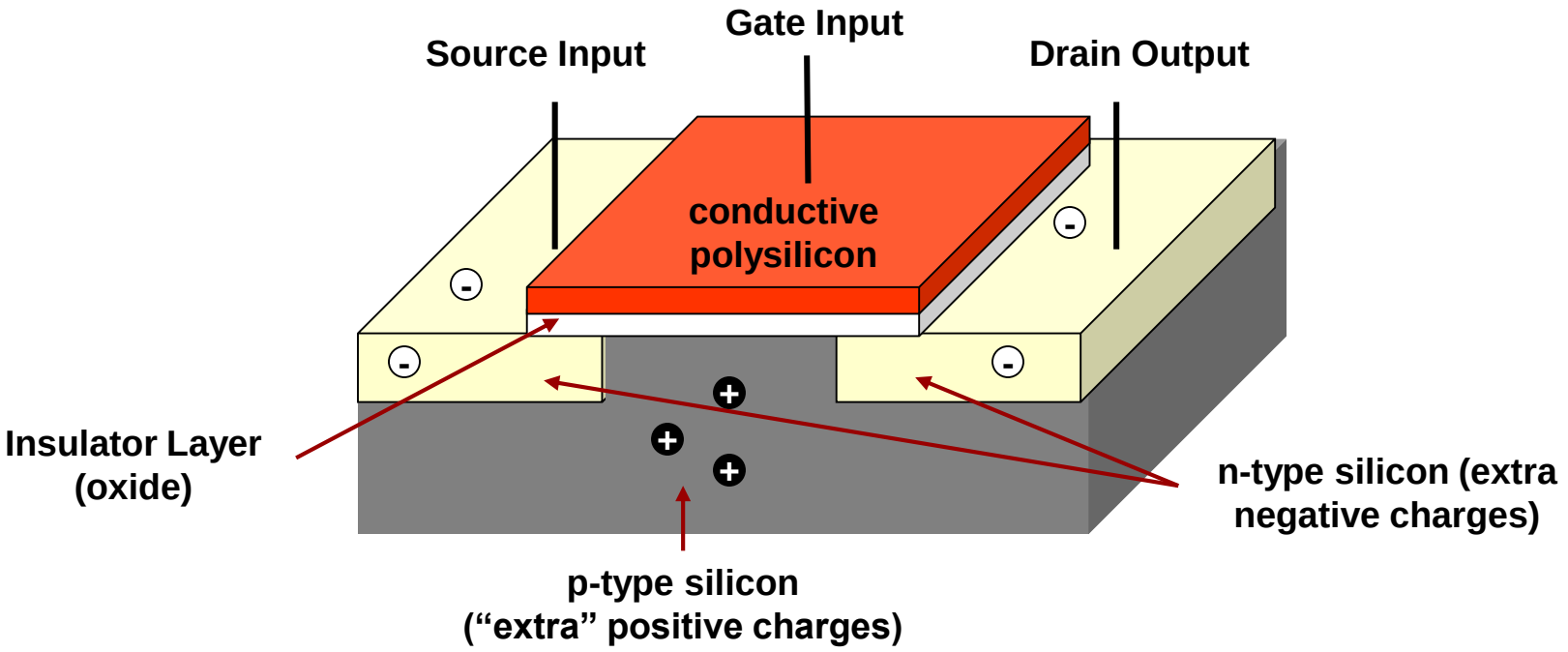
Transistor Physics

- A thin, insulator layer (silicon dioxide or just “oxide”) is placed over the silicon between source and drain



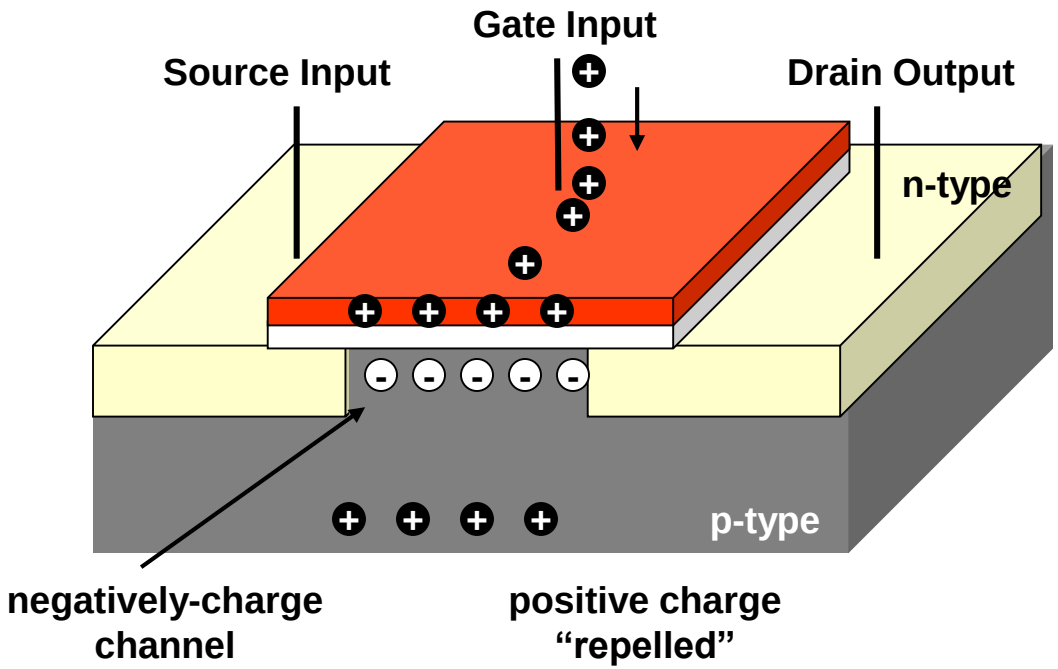
Transistor Physics

- A thin, insulator layer (silicon dioxide or just “oxide”) is placed over the silicon between source and drain
- Conductive polysilicon material is layered over the oxide to form the gate input



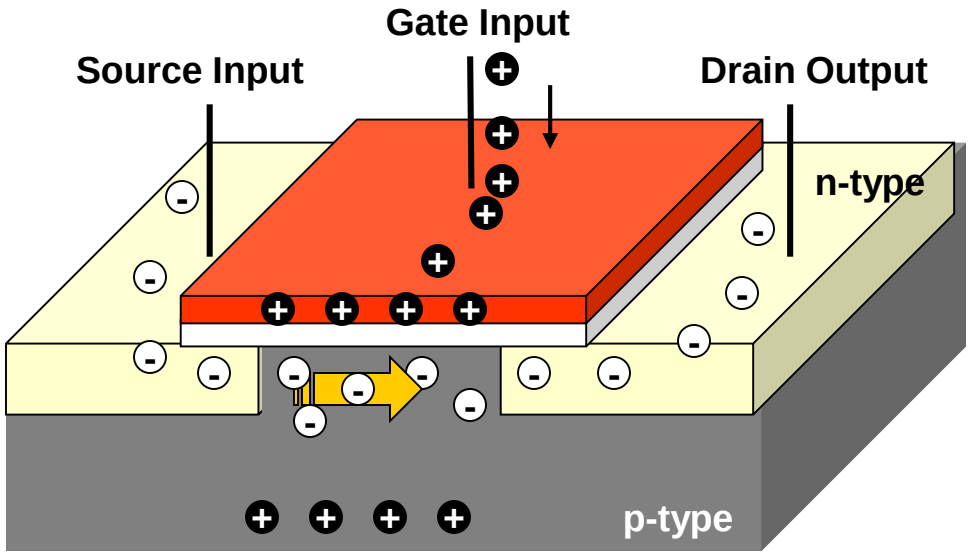
Transistor Physics

- Positive voltage (charge) at the gate input *repels* the extra positive charges in the p-type silicon
- Result is a negative-charge channel between the source input and drain



Transistor Physics

- Electrons can flow through the negative channel from the source input to the drain output
- The transistor is "on"

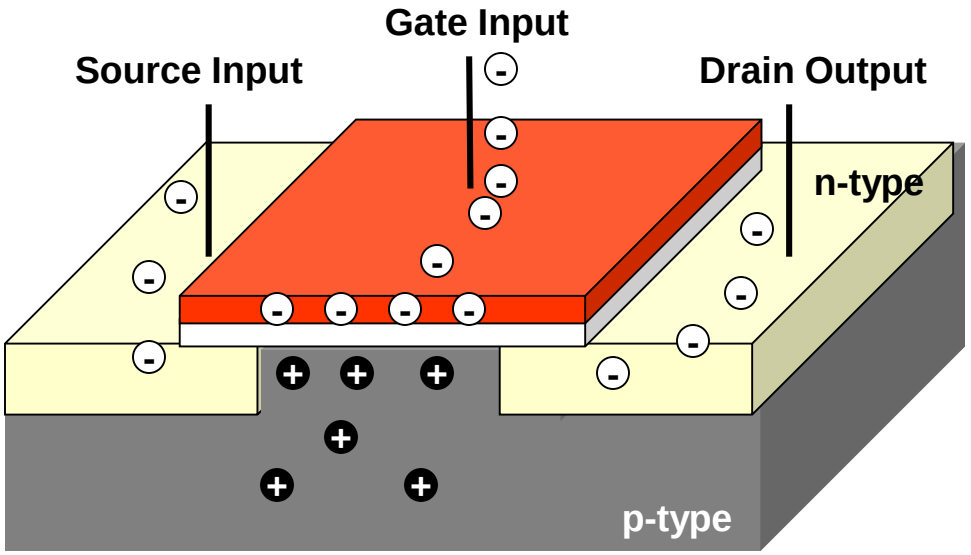


Negative channel between source and drain = Current flow

ON

Transistor Physics

- If a low voltage (negative charge) is placed on the gate, no channel will develop and no current will flow
- The transistor is "off"

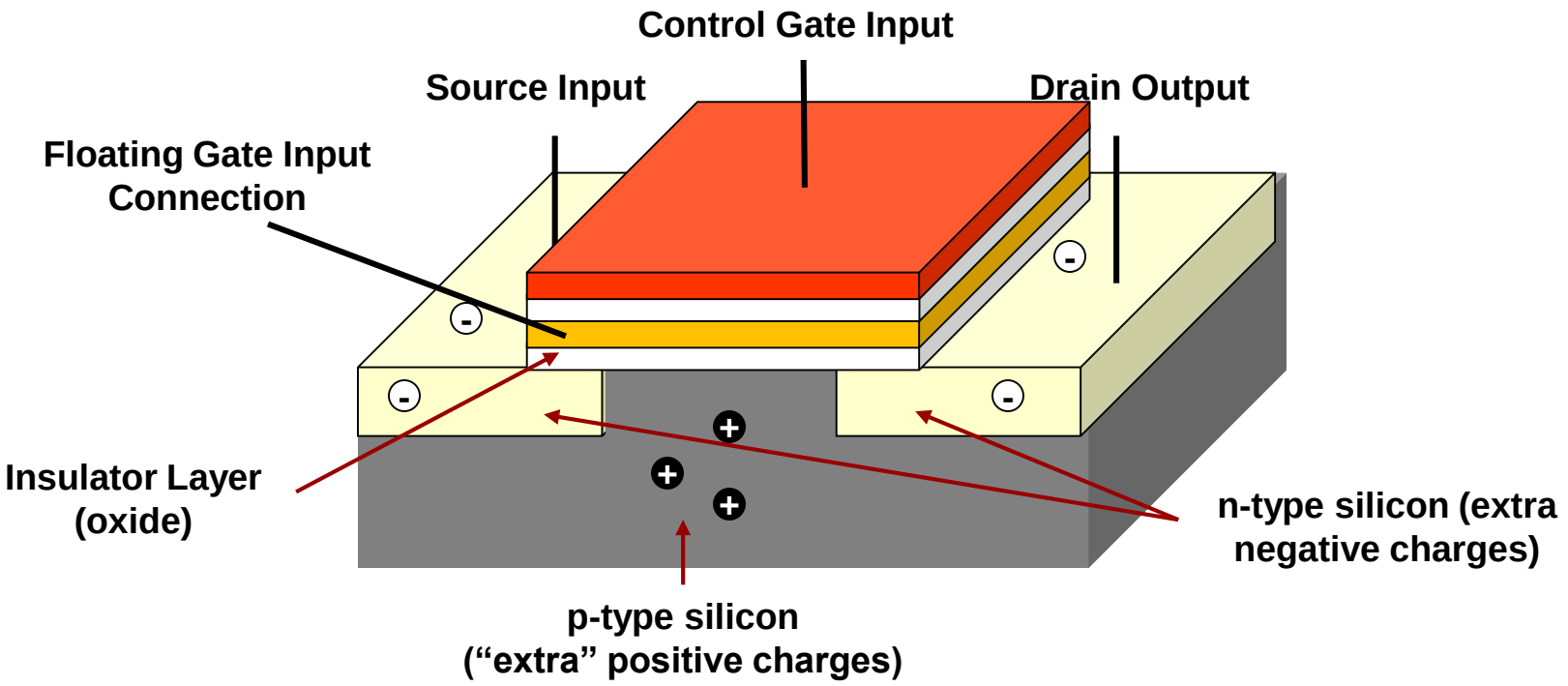


No negative channel between source and drain = No current flow

OFF

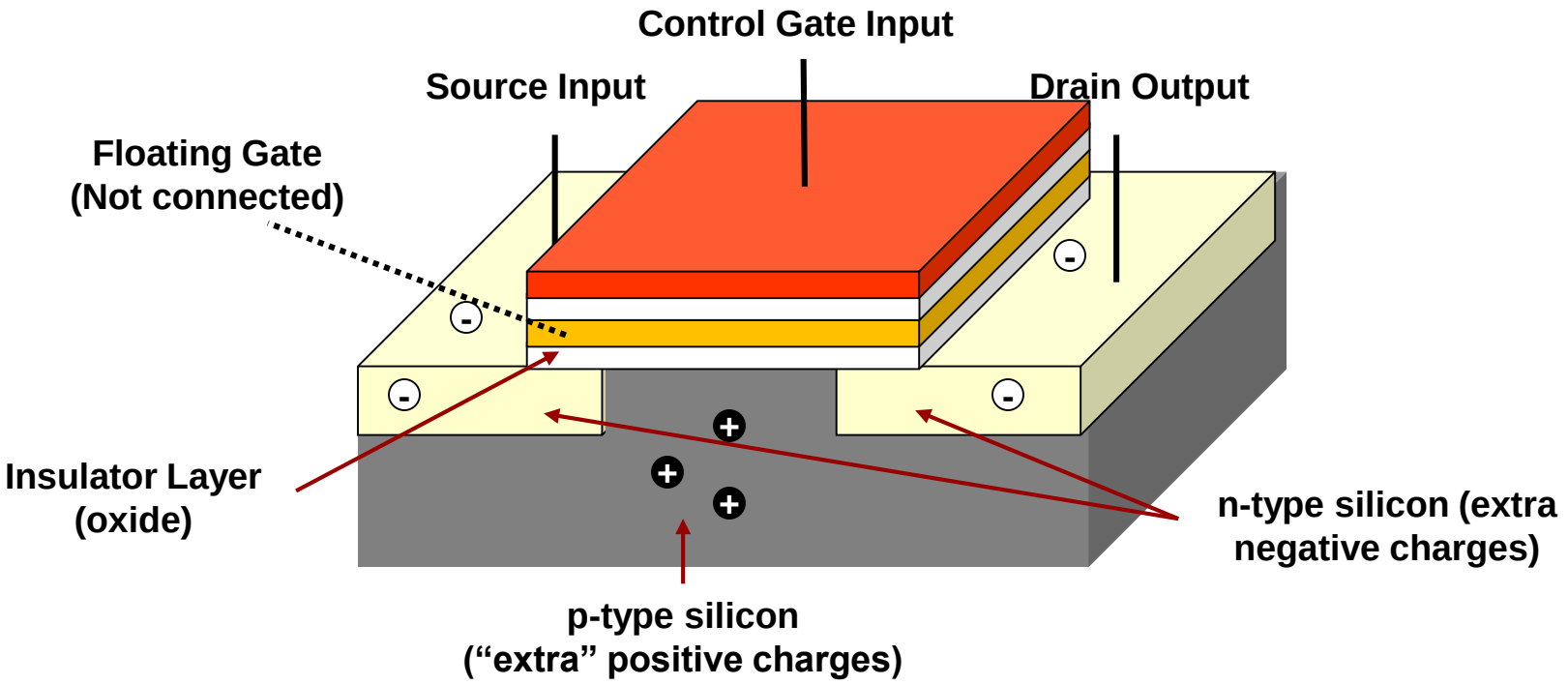
Flash Memory Transistor Physics

- What if we add a second "gate" between the silicon and actual control gate
  - We'll call this the floating gate



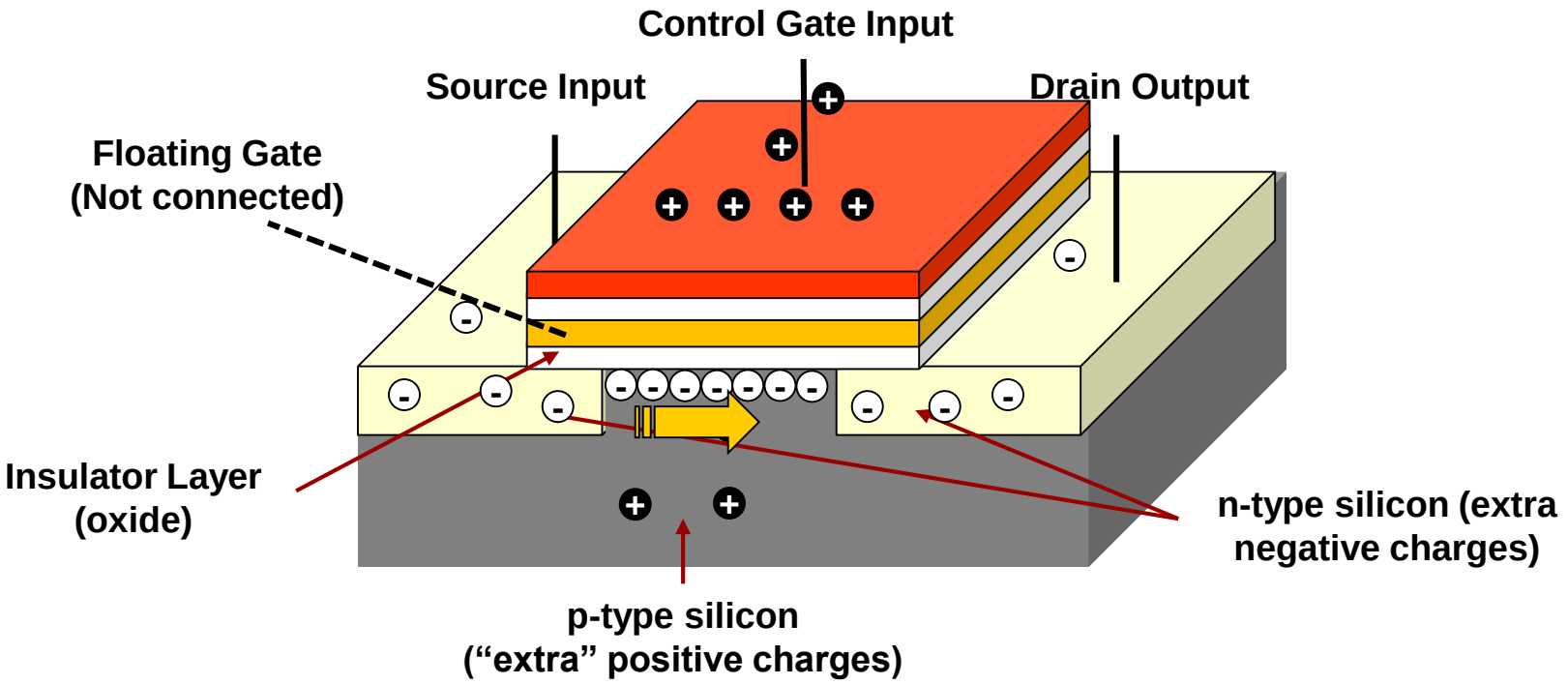
Flash Memory Transistor Physics

- Since it is surrounded by "insulators" any charge we deposit will be trapped and stored (even when power is not applied)



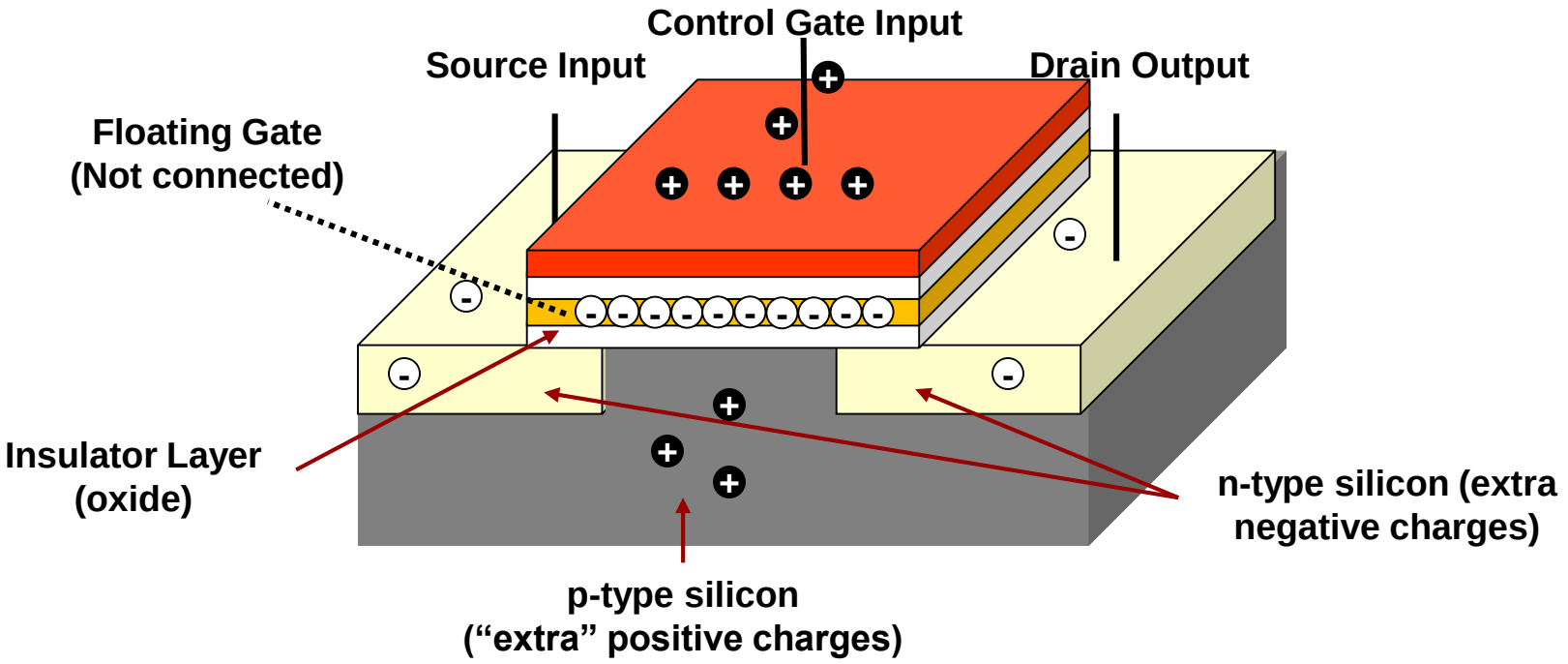
Flash Memory Transistor Physics

- If we have no charge on the floating gate (neutral) then a positive charge on the control gate will still apply an electric field strong enough to create the conductive channel in the underlying silicon and thus turn the transistor ON.



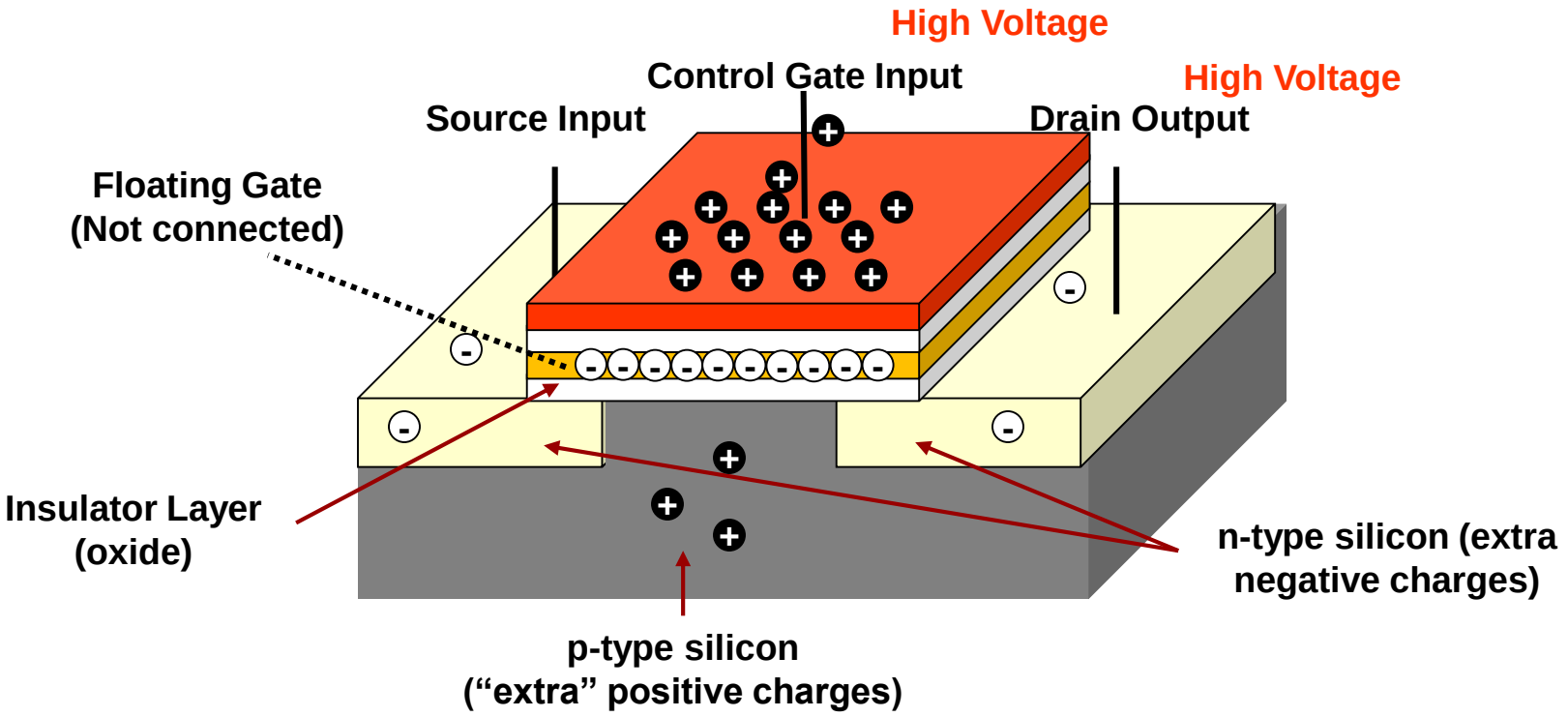
Flash Memory Transistor Physics

- If we trap "negative" charge on the floating gate then no matter what we apply to the control gate the transistor will be OFF



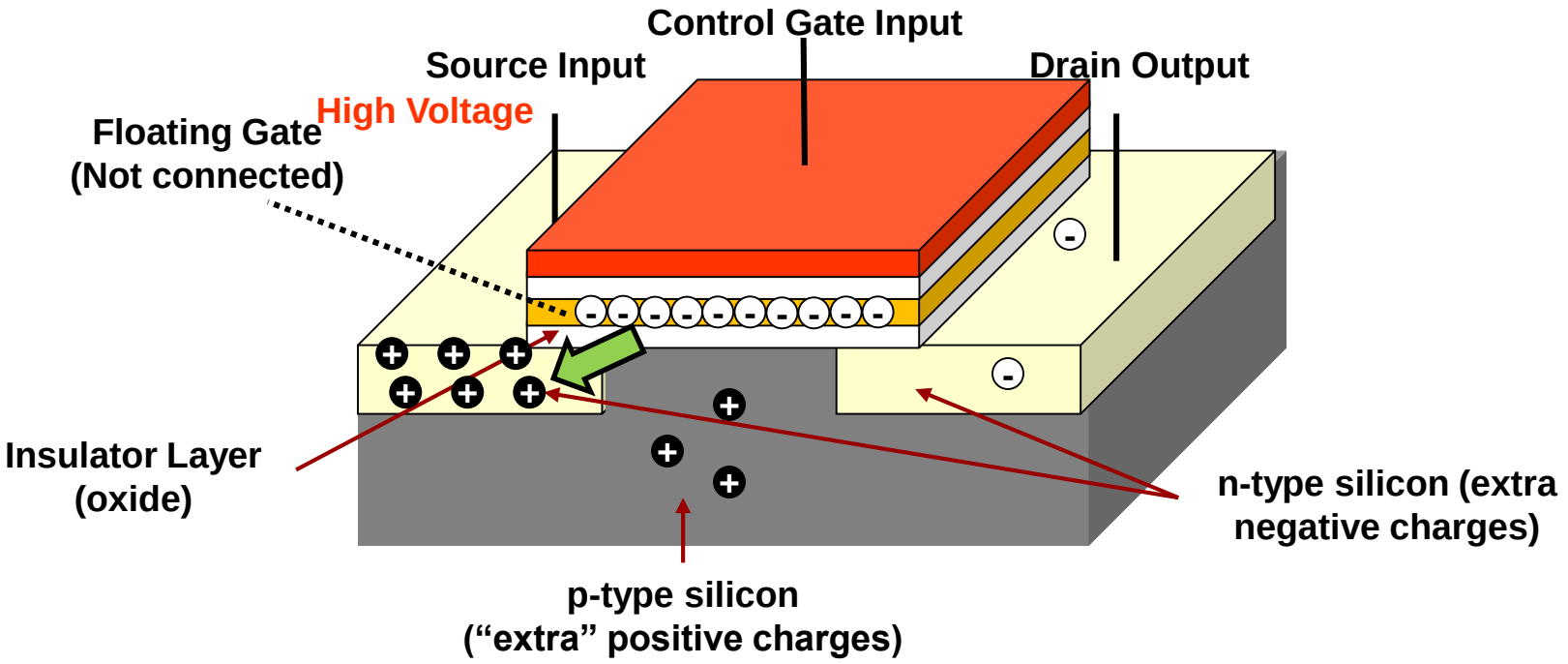
Flash Memory Transistor Physics

- How doe we trap electrons on the FG?
- By Applying a higher than normal voltage to the control gate and drain we can cause "tunneling" of electrons from the source/channel/drain



Flash Memory Transistor Physics

- Erase by apply a high voltage in the opposite polarity (to "suck out" the charge in the FG)



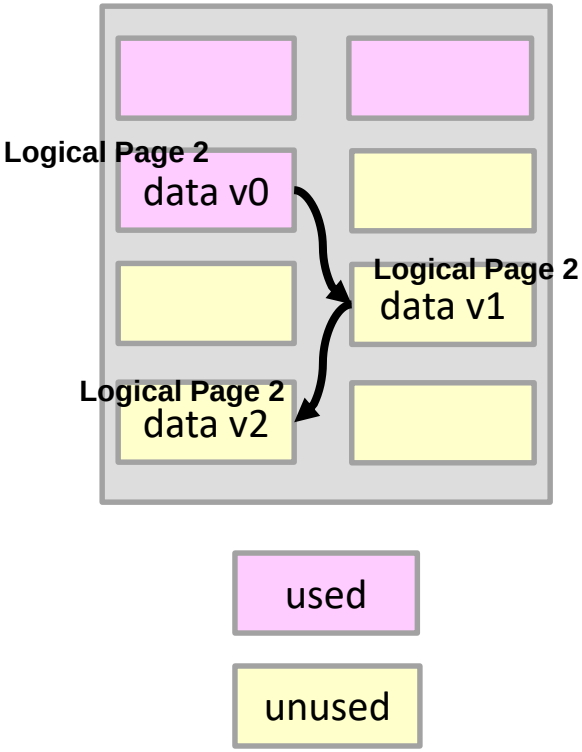
NAND vs. NOR Flash

- 2 Organization Approaches: NAND and NOR Flash
- NOR allows **individual** bytes/words to be read and written (no great speed advantage)
- NAND has increased density but limitations on read/write [Most storage devices use NAND tech.]
- **Erasure** [Both NAND/NOR]: Removal of charge on the FG happens at a block (multi KB chunks) level (aka "erasure block")
  - Due to physical constraints and density reasons (i.e. if we erase in smaller blocks we can't fit as much memory on the chip)
- **Read / Write(Program)**: **Page level**
  - Like a hard drive we must read/write an entire page not individual bits (usually a few microseconds)
- Notice a write from 0101 to 1010 will require erasure

NAND
- **Block (unit of erasure): 128-512KB**
- **Page (unit of reading/writing/programming for NAND): 4KB**

Wear-out & Flash Translation Layer

- All Flash suffers from **wear-out**
  - **After some number of program/erasure cycles (few thousand to few million) the transistor can no longer store its charge reliably**
  - Not only affects reliability but performance since we need to take more countermeasures to deal with the non-working page
- Flash translation layer (FTL)
  - Map logical (external) flash addresses to internal physical locations
  - Rather than erase and re-write a page, simply copy page to a fresh (erased) block and remap the address **[Faster]**
  - Helps spread (even-out) the wearing on cells **[Greater durability]**
  - If a page goes bad, we can just unmap it **[Greater Reliability/Robustness]**
  - **Trim Command:** When a file is deleted, alert the FTL so it can reuse the page



Flash Performance

- Better sequential read throughput
  - HD: 122-204 MB/s
  - SSD: 210-270 MB/s
- MUCH better random read
  - Max latency for single read/write: 75us
  - When many requests present we can overlap and achieve latency of around 26us (1/38500)
- Durability: 1.5PB (PB = 10^{15}) of writes
  - For normal workloads this could last years or decades
  - However if we are constantly writing 200MB/s then the SSD would wear out in about 64 days

Size	
Capacity	300 GB
Page Size	4 KB
Performance	
Bandwidth (Sequential Reads)	270 MB/s
Bandwidth (Sequential Writes)	210 MB/s
Read/Write Latency	75 μs
Random Reads Per Second	38,500
Random Writes Per Second	2,000 2,400 with 20% space reserve
Interface	SATA 3 Gb/s
Endurance	
Endurance	1.1 PB 1.5 PB with 20% space reserve
Power	
Power Consumption Active/Idle	3.7 W / 0.7 W

OS:PP 2nd Ed. Fig. 12.6

Intel 710 SSD specs.

RAID

RAID

- RAID = Redundant Array of Inexpensive Disks
  - Store information redundantly so that if a disk fails the data can be recovered
- Levels
  - RAID 1: Mirror data from one disk on another
    - Can tolerate a disk failure but then must take offline to replace
    - 50% effective storage
  - RAID 5
    - At least 3 disks and store parity
    - Better effective storage
    - Can recreate missing data on the fly if a disk fails and perform a hot-swap with a new disk (no offline penalty)