Well, we'll need at least one MOSFET to serve as the access FET so we can select which bits will be affected by read and write operations.

We can use a simple capacitor for storage, where the value of a stored bit is represented by voltage across the plates of the capacitor.

The resulting circuit is termed a dynamic random-access memory (DRAM) cell.

If the capacitor voltage exceeds a certain threshold, we're storing a "1" bit, otherwise we're storing a "0".

The amount of charge on the capacitor, which determines the speed and reliability of reading the stored value, is proportional to the capacitance.

We can increase the capacitance by increasing the dielectric constant of the insulating layer between the two plates of the capacitor, increasing the area of the plates, or by decreasing the the distance between the plates.

All of these are constantly being improved.

A cross section of a modern DRAM cell is shown here.

The capacitor is formed in a large trench dug into the substrate material of the integrated circuit.

Increasing the depth of the trench will increase the area of the capacitor plates without increasing the cell's area.

The wordline forms the gate of the N-FET access transistor connecting the outer plate of the capacitor to the bitline.

A very thin insulating layer separates the outer plate from the inner plate, which is connected to some reference voltage (shown as GND in this diagram).

You can Google "trench capacitor" to get the latest information on the dimensions and materials used in the construction of the capacitor.

The resulting circuit is quite compact: about 20-times less area/bit than an SRAM bit cell.

There are some challenges however.

There's no circuitry to main the static charge on the capacitor, so stored charge will leak from the outer plate of the capacitor, hence the name "dynamic memory".

The leakage is caused by small picoamp currents through the PN junction with the surrounding substrate, or

subthreshold conduction of the access FET even when it's turned "off".

This limits the amount of time we can leave the capacitor unattended and still expect to read the stored value.

This means we'll have to arrange to read then re-write each bit cell (called a "refresh" cycle) every 10ms or so, adding to the complexity of the DRAM interface circuitry.

DRAM write operations are straightforward: simply turn on the access FET with the wordline and charge or discharge the storage capacitor through the bitline.

Reads are bit more complicated.

First the bitline is precharged to some intermediate voltage, e.g., VDD/2, and then the precharge circuitry is disconnected.

The wordline is activated, connecting the storage capacitor of the selected cell to the bitline causing the charge on the capacitor to be shared with the charge stored by the capacitance of the bitline.

If the value stored by the cell capacitor is a "1", the bitline voltage will increase very slightly (e.g., a few tens of millivolts).

If the stored value is a "0", the bitline voltage will decrease slightly.

Sense amplifiers are used to detect this small voltage change to produce a digital output value.

This means that read operations wipe out the information stored in the bit cell, which must then be rewritten with the detected value at the end of the read operation.

DRAM circuitry is usually organized to have "wide" rows, i.e., multiple consecutive locations are read in a single access.

This particular block of locations is selected by the DRAM row address.

Then the DRAM column address is used to select a particular location from the block to be returned.

If we want to read multiple locations in a single row, then we only need to send a new column address and the DRAM will respond with that location without having to access the bit cells again.

The first access to a row has a long latency, but subsequent accesses to the same row have very low latency.

As we'll see, we'll be able to use fast column accesses to our advantage.

In summary, DRAM bit cells consist of a single access FET connected to a storage capacitor that's cleverly constructed to take up as little area as possible.

DRAMs must rewrite the contents of bit cells after they are read and every cell must be read and written periodically to ensure that the stored charge is refreshed before it's corrupted by leakage currents.

DRAMs have much higher capacities than SRAMs because of the small size of the DRAM bit cells, but the complexity of the DRAM interface circuitry means that the initial access to a row of locations is quite a bit slower than an SRAM access.

However subsequent accesses to the same row happen at speeds close to that of SRAM accesses.

Both SRAMs and DRAMs will store values as long as their circuitry has power.

But if the circuitry is powered down, the stored bits will be lost.

For long-term storage we will need to use non-volatile memory technologies, the topic of the next lecture segment.